



TUGAS AKHIR - KS 141501

RANCANG BANGUN FITUR PENGELOLAAN PENYARINGAN PRODUK PADA SITUS PENJUALAN PRODUK PERANGKAT LUNAK UNTUK MEMPERMUDAH PENCARIAN

DEVELOPMENT OF PRODUCT FILTER MANAGEMENT FEATURE IN SOFTWARE SALES SITES TO SIMPLIFY SEARCH

UMAR AL AQSHO
NRP 5213 100 107

Dosen Pembimbing :
Rully Agus Hendrawan, S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KS 141501

RANCANG BANGUN FITUR PENGELOLAAN PENYARINGAN PRODUK PADA SITUS PENJUALAN PRODUK PERANGKAT LUNAK UNTUK MEMPERMUDAH PENCARIAN

**UMAR AL AQSHO
NRP 5213 100 107**

Dosen Pembimbing :
Rully Agus Hendrawan, S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 141501

DEVELOPMENT OF PRODUCT FILTER MANAGEMENT FEATURE IN SOFTWARE SALES SITES TO SIMPLIFY SEARCH

UMAR AL AQSHO
NRP 5213 100 107

SUPERVISOR:
Rully Agus Hendrawan, S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

LEMBAR PENGESAHAN

RANCANG BANGUN FITUR PENGELOLAAN PENYARINGAN PRODUK PADA SITUS PENJUALAN PRODUK PERANGKAT LUNAK UNTUK MEMPERMUDAH PENCARIAN

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

UMAR AL AQSHO
NRP. 5213 100 107

Surabaya, 14 Juli 2017

KEPALA DEPARTEMEN SISTEM INFORMASI



Dr. Ir. Aris Tjahyanto, M.Kom
NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN
RANCANG BANGUN FITUR PENGELOLAAN
PENYARINGAN PRODUK PADA SITUS PENJUALAN
PRODUK PERANGKAT LUNAK UNTUK
MEMPERMUDAH PENCARIAN


TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Oleh :


UMAR AL AQSHO
NRP. 5213 100 107

Disetujui Tim Penguji : Tanggal Ujian : 7 Juli 2017
Periode Wisuda : September 2017

Rully Agus Hendrawan, S.Kom, M.Eng


(Pembimbing I)

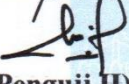
Andre Parvian Aristio, S.Kom., M.Sc.


(Pembimbing II)

Erma Suryani, ST, MT, Ph.D


(Penguji I)

Mahendrawati ER, ST, M.Sc, Ph.D


(Penguji II)

RANCANG BANGUN FITUR PENGELOLAAN PENYARINGAN PRODUK PADA SITUS PENJUALAN PRODUK PERANGKAT LUNAK UNTUK MEMPERMUDAH PENCARIAN

Nama Mahasiswa : Umar Al Aqsho
NRP : 5213 100 107
Jurusan : Sistem Informasi FTIf-ITS
Pembimbing 1 : Rully Agus Hendrawan, S.Kom., M.Eng.
Pembimbing 2 : Andre Parvian Aristio, S.Kom., M.Sc.

ABSTRAK

Produk merupakan informasi yang dicari oleh pengguna, pengguna dapat berbagi, berkomentar, berlangganan, menyumbangkan, mengikuti atau membeli dari produk yang tersebar secara online.

Untuk memudahkan pencarian produk yang bernilai serta sesuai dengan kebutuhan pengguna hanya dengan menggunakan kata kunci tidaklah cukup, pengguna atau pemilik mobile device membutuhkan cara yang mudah untuk menemukan aplikasi sesuai kebutuhan dirinya. Saat ini, pengguna perlu memikirkan atau melakukan brainstorm keyword jika ingin mencari produk yang dibutuhkan. Namun, sering kali keyword untuk mencari produk yang dibutuhkan tersebut tidak terpikirkan dan keyword harus terus menerus dipertajam. Sehingga pengguna butuh usaha lebih untuk memahami kebutuhan dirinya sendiri. Pengguna juga tidak dapat menemukan produk yang mereka butuhkan dikarenakan salah memasukan keyword pencarian, meskipun sebenarnya produk/solusi tersedia.

Pada Penelitian ini, dibangun prototipe yang memiliki fungsi pencarian dan pengelolaan penyaringan berdasarkan kebutuhan pengguna dengan memberikan label terhadap aplikasi yang saling terkait serta menerapkan fitur filter search.

Filter search mengacu kepada: teknik untuk mengakses informasi yang telah ditata berdasarkan klasifikasi, yang memungkinkan pengguna untuk menjelajahi kumpulan informasi dengan menerapkan beberapa nilai terhadap pencariannya. Filter search dapat membantu dalam melakukan penyempitan pencarian sehingga produk yang diinginkan dapat lebih mudah ditemukan.

Pembangunan prototipe dengan tujuan mendapatkan umpan balik sehingga dapat menciptakan hasil akhir yang baik serta dapat dilakukan pengujian kepada pengguna secara langsung mengenai pencarian dan penyaringan produk, dengan pengujian dapat diketahui bahwa produk yang pengguna cari dan pilih sesuai dengan yang dibutuhkan oleh mereka.

Hasil dari tugas akhir ini adalah sebuah aplikasi web berbasis PHP dengan halaman untuk mencari produk aplikasi Google Play dengan beberapa pilihan penyaringan dan halaman untuk menampilkan hasil pencarian produk. Dihasilkan juga penyimpanan pencarian dan penyaringan yang telah dilakukan oleh pengguna menjadi sebuah template pencarian. Dengan diciptakannya prototipe aplikasi ini didapatkan 71,25% kemudahan penggunaan yang menunjukkan prototipe dapat digunakan untuk memudahkan pengguna dalam mencari produk aplikasi yang dibutuhkan.

Kata Kunci : Produk, Pengelolaan Penyaringan, Filter Search, Prototipe, Penyaringan

DEVELOPMENT OF PRODUCT FILTER MANAGEMENT FEATURE IN SOFTWARE SALES SITES TO SIMPLIFY SEARCH

Student Name : Umar Al Aqsho
Student ID : 5213 100 107
Department : Sistem Informasi FTIf-ITS
Supervisor 1 : Rully Agus Hendrawan, S.Kom., M.Eng.
Supervisor 2 : Andre Parvian Aristio, S.Kom., M.Sc.

ABSTRACT

Products are information sought by users, users can share, comment, subscribe, donate, follow or buy from products spread online.

In order to facilitate the search for valuable products as well as to suit the needs of users simply by using keywords is not enough, the users or owners of mobile devices need an easy way to find the app as per their needs. Currently, users need to think or brainstorm keywords if they want to find the product they need. However, often keywords to find the required product is unthinkable and the keyword must be continuously refined. So users need more effort to understand their own needs. Users also can not find the product they need due to incorrectly entering the search keyword, even though the actual product/solution is available.

In this research, we built prototype which has search function and filtering management based on user requirement by labeling interrelated application and applying filter search feature.

Filter search refers to: techniques for accessing information that has been organized by classification, allowing users to navigate through a collection of information by applying some value to its search. Filter search can help in narrowing the search so that the desired product can be more easily found.

Development of prototypes with the goal of getting reverse so as to create a good end result and can be tested to the user directly about the search and filtering of products, by testing it can be seen that the products that users search and select in accordance with that required by them.

The result of this final project is a PHP-based web application with a page to search Google Play app products with multiple filtering options and pages to display product search results. Also generated search storage and filtering has been done by the user into a search template. With the creation of this prototype application obtained 71.25 ease of use that shows the prototype can be used to facilitate users in finding the required application products.

Keywords : Produk, Pengelolaan Penyaringan, Filter Search, Prototipe, Penyaringan

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, karena dengan limpahan rahmat-Nya penulis dapat menyelesaikan laporan Tugas Akhir yaitu **Rancang Bangun Fitur Pengelolaan Penyaringan Produk Pada Situs Penjualan Produk Perangkat Lunak Untuk Mempermudah Pencarian**. Tak lupa penulis juga mengucapkan terima kasih sebanyak-banyaknya kepada semua pihak yang sudah berkenan membantu. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT yang senantiasa melimpahkan berkah dan rahmat-Nya selama penulis mengerjakan Tugas Akhir.
2. Keluarga penulis, khususnya Ummi, Abi, serta Adik-adik yang senantiasa mendoakan kelancaran serta kesuksesan dalam pengerjaan Tugas Akhir.
3. Bapak Ir. Aris Tjahyanto, M.Kom. selaku Ketua Jurusan Sistem Informasi ITS Surabaya.
4. Bapak Rully Agus Hendrawan, S.Kom., M.Eng selaku Dosen Pembimbing I dan Ketua Laboratorium Sistem Enterprise yang telah membimbing dan bersedia memberikan fasilitas tempat untuk mengerjakan Tugas Akhir ini.
5. Bapak Andre Parvian Aristio, S.Kom., M.Sc selaku Dosen Pembimbing II yang telah memberikan motivasi dan membimbing sehingga saya terus mengusahakan yang terbaik selama mengerjakan Tugas Akhir ini.
6. Ibu Mahendrawati ER, ST, M.Sc, Ph.D selaku Dosen Wali yang telah memberikan banyak masukan, motivasi, membimbing, dan tempat penulis berkeluh kesah selama kuliah sejak tahun pertama hingga penulis mengerjakan Tugas Akhir ini.

7. Responden yang dengan sukarela membantu memberikan *feedback* dan seluruh masukkan sehingga penulis dapat melakukan pengembangan aplikasi.
8. Teman-teman Laboratorium SE dan ADDI yang telah mempersilahkan penulis bernaung dan mencari inspirasi dalam mengerjakan Tugas Akhir ini.
9. Seluruh teman – teman, Keluarga Beltranis yang selalu memberikan banyak dukungan, pengalaman, cerita, dan kenangan selama penulis melakukan penelitian.
10. Mas dan Mbak serta adik – adik jurusan Sistem Informasi yang telah memberikan semangat serta bantuan kepada penulis.
11. Seluruh civitas akademika Jurusan Sistem Informasi ITS atas dukungan yang diberikan selama pengerjaan Tugas Akhir.
12. Dan seluruh pihak yang telah membantu penulis baik secara langsung maupun tidak langsung dan telah memberikan dukungan sehingga Tugas Akhir ini dapat terselesaikan dengan baik.

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan karena kesempurnaan sejatinya hanya milik Allah SWT, maka saran dan kritik yang konstruktif dari semua pihak sangat diharapkan demi perbaikan selanjutnya.

Surabaya, 16 Juni 2017

Penulis

DAFTAR ISI

ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN	1
1.1. Latar belakang.....	1
1.2. Perumusan masalah.....	2
1.3. Batasan masalah.....	3
1.4. Tujuan tugas akhir.....	3
1.5. Manfaat tugas akhir.....	3
1.6. Relevansi tugas akhir	4
BAB II TINJAUAN PUSTAKA	5
2.1. Penelitian sebelumnya.....	5
2.2. Dasar teori.....	7
2.2.1. Google play api	8
2.2.2. Search in relational database	10
2.2.3. Laravel	11
2.2.4. Prototyping.....	11
2.2.5. Usability testing.....	12
BAB III METODOLOGI.....	13
3.1. Tahapan	14
3.1.1. Studi literatur.....	14
3.1.2. Pencarian kebutuhan.....	14
3.1.3. Perancangan prototipe	14
3.1.4. Pengujian aplikasi	15
3.1.5. Hasil dan pembahasan	15
BAB IV ANALISIS DAN PERANCANGAN	17
4.1. Pencarian kebutuhan	17
4.1.1 Wawancara kebutuhan	17
4.2. Perancangan pengambilan data	18

4.2.1.	Data yang akan diambil	18
4.2.2.	Desain basis data	24
4.2.3.	Rancang bangun prototipe	43
4.2.4.	Antar muka pengguna.....	44
4.2.5.	Use case.....	46
4.3.	Rancangan perbaikan usabilitas fitur pengelolaan.....	49
4.3.1.	Pemberian nama template secara otomatis	49
4.3.2.	Pengubahan nama template pencarian	50
4.3.3.	Perancangan pencarian judul template.....	52
BAB V	IMPLEMENTASI	53
5.1.	Lingkungan implementasi	53
5.2.	Struktur direktori.....	53
5.3.	Pengambilan data	54
5.3.1.	Pembuatan scraper.....	55
5.4.	Rancang bangun prototipe.....	61
5.4.1.	Implementasi halaman pencarian.....	61
5.4.2.	Implementasi halaman hasil pencarian	73
5.4.3.	Implementasi perbaikan usabilitas fitur pengelolaan.....	86
BAB VI	HASIL DAN PEMBAHASAN	89
6.1.	Hasil prototipe aplikasi	89
6.1.1.	Halaman pencarian	89
6.1.2.	Halaman hasil pencarian.....	90
6.1.3.	Hasil pembuatan penyaringan.....	90
6.2.	Hasil perbaikan usabilitas fitur pengelolaan	95
6.3.	Hasil uji usabilitas	96
BAB VII	KESIMPULAN DAN SARAN.....	101
7.1.	Kesimpulan	101
7.2.	Saran	102
DAFTAR PUSATAKA.....		103
BIODATA PENULIS.....		105
LAMPIRAN 1 SOURCE CODE APLIKASI.....		107
LAMPIRAN 2 USABILITY TEST PLAN DASHBOARD		112
LAMPIRAN 3 HASIL WAWANCARA.....		113
LAMPIRAN 4 TEST CASE		114
LAMPIRAN 5 CONSENT FORM.....		127
LAMPIRAN 6 SYSTEM USABILITY SCALE FORM		129

DAFTAR GAMBAR

Gambar 1.1 Kerangka kerja riset laboratorium sistem enterprise.....	4
Gambar 2.1. Bagan dasar teori	8
Gambar 4.1 Metode penelitian	13
Gambar 5.1 Alur pengambilan data sebelum dimodifikasi dari google play	19
Gambar 5.2 Alur pengambilan data dari api + json	24
Gambar 5.3 Entiti yang saling berkaitan	25
Gambar 5.4 Identifikasi relasi antara entiti.....	25
Gambar 5.5 Menghubungkan atribut entiti.....	26
Gambar 5.6 Domain atribut Googleplay_scrap_list dan Googleplay_scrap_app.....	27
Gambar 5.7 Domain atribut application_facet dan Google_scrap_app.....	27
Gambar 5.8 Domain atribut applications_facet dan applications_fact	28
Gambar 5.9 melakukan generalisasi menjadi tabel applications	29
Gambar 5.10 Entitas dan atribut Google_scrap_app	36
Gambar 5.11 Bentuk MRD dari Googleplay_scrap_list	36
Gambar 5.12 Entitas dan atribut Googleplay_scrap_app	36
Gambar 5.13 Bentuk MRD dari Googleplay_scrap_app	37
Gambar 5.14 Entitas dan atribut applications	37
Gambar 5.15 Bentuk MRD dari applications	38
Gambar 5.16 Entitas dan atribut applications_facet	38
Gambar 5.17 Bentuk MRD dari applications_facets	38
Gambar 5.18 Entitas dan atribut yang dimiliki	39
Gambar 5.19 Bentuk MRD dari applications_fact.....	39
Gambar 5.20 Weak entity Googleplay_scrap_list dengan Googleplay_scrap_app	40
Gambar 5.21 Relasi Googleplay_scrap_list dengan Googleplay_scrap_app dalam bentuk MRD.....	40
Gambar 5.22 Relasi one-to-one Googleplay_scrap_list dengan Googleplay_scrap_app	41

Gambar 5.23 Relasi pada model data fisikal	43
Gambar 5.24 Halaman pencarian	44
Gambar 5.25 Halaman hasil pencarian	45
Gambar 5.26 Use case diagram oleh pengguna	46
Gambar 5.27 Pemberian nama template secara otomatis.....	49
Gambar 5.28 Notifikasi pemberian judul template	50
Gambar 5.29 Pengubahan nama template pencarian	50
Gambar 5.30 Mengubah nama/judul pencarian template.....	51
Gambar 5.31 Notifikasi pemberian judul template	51
Gambar 5.32 Perancangan pencarian judul template	52
Gambar 6.1 Struktur direktori aplikasi	54
Gambar 6.2 Konfigurasi koneksi ke basis data	55
Gambar 6.3 Memasukan data ke proses cURL	56
Gambar 6.4 Pengolahan data dari cURL	58
Gambar 6.5 Memasukan data dari proses list	59
Gambar 6.6 Mengambil data yang belum terambil dari proses list.....	59
Gambar 6.7 Mengkonversi data yang didapat dari proses app	60
Gambar 6.8 Memasukan data yang sudah dikonversi ke basis data.....	60
Gambar 6.9 Template dan kolom pencarian untuk halaman pencarian	62
Gambar 6.10 Penyaringan range slider untuk price produk ...	62
Gambar 6.11 Penyaringan range slider untuk rating produk..	63
Gambar 6.12 Penyaringan pilihan produk berdasarkan usia ..	64
Gambar 6.13 Penyaringan pilihan produk berdasarkan kategori produk	68
Gambar 6.14 Penyaringan pilihan produk berdasarkan versi android	70
Gambar 6.15 Penyaringan pilihan produk yang dapat diakses secara early	70
Gambar 6.16 Penyaringan pilihan produk yang memiliki iklan dalam aplikasi.....	70
Gambar 6.17 Query pemrosesan pencarian dengan template	72
Gambar 6.18 Query pemrosesan pencarian dengan kolom..	72

Gambar 6.19 Query untuk memproses data saat tidak dilakukan pencarian	73
Gambar 6.20 Menampilkan hasil pencarian	74
Gambar 6.21 Javascript dan ajax untuk menampilkan hasil penyaringan dengan range slider.	77
Gambar 6.22 Query untuk memproses penyaringan range slider rating	78
Gambar 6.23 Query untuk memproses penyaringan range slider rating	79
Gambar 6.24 Masukan untuk menyimpan template pencarian	84
Gambar 6.25 Query untuk menyimpan masukan sebagai template pencarian kedalam basis data.....	85
Gambar 6.26 kode pemberian nama template secara otomatis	86
Gambar 7.1 Halaman pencarian produk	89
Gambar 7.2 Halaman hasil pencarian.....	90
Gambar 7.3 Hasil pemberian nama template secara otomatis	96
Gambar 7.4 Hasil rekapitulasi SUS	99
Gambar 10.1 Routing dari sistem yang dikembangkan	107
Gambar 10.2 dropdown template dan kolom input kata kunci	107
Gambar 10.3 kolom input judul template dan tombol simpan	107
Gambar 10.4 filter checkbox yang tersedia pada halaman pencarian.....	108
Gambar 10.5 filter range slider yang tersedia pada halaman pencarian.....	109
Gambar 10.6 Menampilkan hasil pencarian	110
Gambar 10.7 Query filter rating	111
Gambar 10.8 Query filter harga.....	111
Gambar 10.9 Query pencarian dengan template dropdown .	111
Gambar 10.10 Query pencarian dengan kata kunci dan filter checkbox	111

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Hasil kajian penelitian sebelumnya	5
Tabel 2.2 Hasil kajian penelitian sebelumnya	6
Tabel 5.1 Kebutuhan pengguna	18
Tabel 5.2 Deskripsi data atribut tabel applications	19
Tabel 5.3 Deskripsi data atribut tabel applications_facet	22
Tabel 5.4 Deskripsi data atribut tabel applications_fact	22
Tabel 5.5 Data dictionary tabel applications	31
Tabel 5.6 Data dictionary tabel applications_facets	34
Tabel 5.7 Data dictionary tabel applications_fact	34
Tabel 5.8 Daftar use case	46
Tabel 5.9 UCP-1: Memilih template pencarian	47
Tabel 5.10 UCP-2: Mencari produk aplikasi	47
Tabel 5.11 UCP-3: Melihat hasil aplikasi	47
Tabel 5.12 UCP-4: Memasukan kata pencarian	48
Tabel 5.13 UCP-5: Menambah filter	48
Tabel 5.14 UCP-6: Mengubah filter	48
Tabel 5.15 UCP-7: Menyimpan filter dan katakunci pencarian menjadi template	48
Tabel 6.1 Lingkungan implementasi (perangkat keras)	53
Tabel 6.2 Lingkungan implementasi (perangkat lunak)	53
Tabel 7.1 Metadata yang dapat dijadikan penyaringan	91
Tabel 7.2 Hasil meta data menjadi penyaringan	95
Tabel 7.3 Backlog dari pengujian aplikasi	96
Tabel 7.4 Test Case pengujian aplikasi	97
Tabel 7.5 Pernyataan dan justifikasi SUS	98
Tabel 7.6 Perhitungan skor pengujian usabilitas	100

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Bagian ini akan menjelaskan tahap mengidentifikasi permasalahan yang diangkat pada Tugas Akhir, dimulai dari latar belakang permasalahan, perumusan permasalahan, batasan masalah, tujuan serta manfaat dari pengerjaan Tugas Akhir tersebut.

1.1. Latar belakang

Produk merupakan informasi yang dicari oleh pengguna, pengguna dapat berbagi, berkomentar, berlangganan, menyumbangkan, mengikuti atau membeli dari produk yang tersebar secara *online* [1].

Namun, saat mencari produk pengguna dihadapi masalah dengan hasil pencarian mana yang paling cocok dengan kebutuhan pengguna/pemilik *mobile device*. Hasil pencarian yang ada saat ini biasanya didapatkan berdasarkan kesamaan/kemiripan teks atau kata kunci, bukan berdasarkan kebutuhan yang pengguna inginkan. Maka dibutuhkan pencarian dan penyaringan yang sesuai dengan kebutuhan untuk membantu pengguna mendapatkan produk yang tepat bagi mereka.

Filter search mengacu kepada: teknik untuk mengakses informasi yang telah ditata berdasarkan klasifikasi, yang memungkinkan pengguna untuk menjelajahi kumpulan informasi dengan menerapkan beberapa filter terhadap pencariannya [2]. Dengan menerapkan *filter search* pengguna bisa mendapatkan pilihan kondisi yang terkait dengan pencariannya sehingga pengguna dapat menemukan produk yang sesuai [3].

Pembangunan prototipe dengan tujuan mendapatkan umpan balik sehingga dapat menciptakan hasil akhir yang baik serta dapat dilakukan pengujian kepada pengguna secara langsung mengenai pencarian dan penyaringan produk, dengan pengujian

dapat diketahui bahwa produk yang pengguna cari dan pilih sesuai dengan yang dibutuhkan oleh mereka [4].

Pada era perdagangan yang semakin berkembang ini, para penjual ingin memiliki katalog produk yang besar dan terlengkap sehingga konten yang dimiliki oleh penjual menjadi semakin ramai. Namun, pengguna saat ini lebih memilih untuk mencari dan membeli produk yang mereka butuhkan, tetapi banyak yang merasakan kebingungan dengan banyaknya pilihan yang ditawarkan, sehingga pengguna ditantang untuk menyaring dan mencari produk yang berkualitas [5]. Ketika hasil tidak ditemukan pengguna perlu terus menerus mempertajam kata kunci pencarian untuk menemukan produk yang dibutuhkan.

Hasil dari tugas akhir ini adalah sebuah aplikasi web berbasis PHP/HTML yang dapat membantu pencarian dan penyaringan konteks, dengan adanya aplikasi ini diharapkan dapat digunakan untuk meningkatkan paparan produk kepada konsumen sehingga dapat membantu peningkatan penjualan serta mempercepat untuk menemukan produk yang sesuai dengan kebutuhan diri pengguna.

1.2. Perumusan masalah

Berdasarkan latar belakang yang telah dipaparkan pada bagian sebelumnya, maka rumusan masalah yang akan diselesaikan pada Tugas Akhir ini adalah sebagai berikut :

Pengguna *mobile device* membutuhkan cara yang mudah untuk menemukan aplikasi sesuai kebutuhan dirinya. Saat ini, pengguna perlu memikirkan atau melakukan *brainstorm keyword* jika ingin mencari produk yang dibutuhkan. Namun, sering kali *keyword* untuk mencari produk yang dibutuhkan tersebut tidak terpikirkan dan *keyword* harus terus-menerus dipertajam. Sehingga pengguna butuh usaha lebih untuk memahami kebutuhan dirinya sendiri. Pengguna tidak dapat menemukan produk yang mereka butuhkan dikarenakan salah

memasukan *keyword* pencarian, meskipun sebenarnya produk/solusi tersedia.

1.3. Batasan masalah

Batasan masalah yang dapat diperhatikan dalam penyelesaian Tugas Akhir ini adalah sebagai berikut :

1. Sumber data yang digunakan dalam aplikasi yang dirancang didapatkan hanya dari Google Play sejumlah 500 produk aplikasi.
2. Partisipan pengujian aplikasi yang dibangun hanya terbatas sebanyak 10 orang.
3. Proses *Tagging* produk tiap aplikasi hanya dilakukan oleh penulis dan 10 partisipan.

1.4. Tujuan tugas akhir

Tujuan utama dari pembuatan tugas akhir adalah:

1. Merancang model data logikal terkait atribut produk dan produk katalog.
2. Membangun aplikasi web berbasis PHP/HTML yang dapat membantu pencarian dan penyaringan produk.
3. Menganalisis usabilitas sistem dalam fitur pengelolaan penyaringan produk dari sisi pengguna.

1.5. Manfaat tugas akhir

Berikut manfaat yang diperoleh, dengan melihat dari dua belah sudut pandang, yaitu sudut pandang pemilik situs penjualan perangkat lunak dan pihak konsumen:

Bagi pemilik situs penjualan perangkat lunak

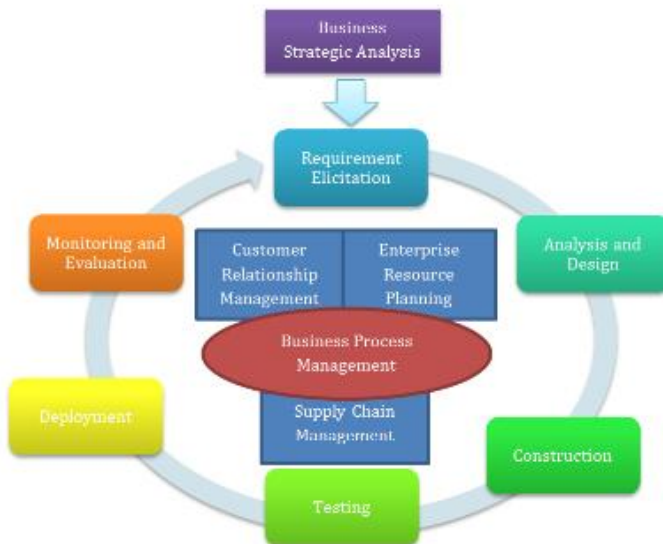
Meningkatkan paparan produk kepada konsumen sehingga dapat membantu peningkatan penjualan.

Bagi konsumen

Mempercepat untuk menemukan produk yang sesuai dengan kebutuhan.

1.6. Relevansi tugas akhir

Laboratorium Sistem Enterprise (SE) Jurusan Sistem Informasi ITS memiliki empat topik utama, yaitu customer relationship management (CRM), enterprise resource planning (ERP), supply chain management (SCM) dan business process management (BPM) seperti yang terdapat pada Gambar 1.1. Dalam tugas akhir yang dikerjakan oleh penulis mengambil customer relationship management (CRM) sebagai topik utama. Mata kuliah yang berkaitan dengan CRM adalah Manajemen Rantai Pasok dan Hubungan Pelanggan (MRPHP).



Gambar 1.1 Kerangka kerja riset laboratorium sistem enterprise

BAB II

TINJAUAN PUSTAKA

Bagian ini akan memberikan penjelasan mengenai penelitian maupun studi literatur sebelumnya yang berkaitan dan dijadikan sebagai acuan selama pengerjaan tugas akhir, serta landasan teori yang berkaitan dengan tugas akhir yang dapat membantu pemahaman selama pengerjaan tugas akhir ini.

2.1. Penelitian sebelumnya

Selama pengerjaan tugas akhir ini, terdapat beberapa penelitian yang telah dilakukan sebelumnya yang dapat dijadikan sebagai bahan kajian maupun referensi untuk studi literatur. Penelitian tersebut lalu dikaji untuk dilihat dari gambaran umum, tujuan hasil, dan keterkaitannya dengan penelitian tugas akhir ini. Hasil dari kajian tersebut dapat kita lihat pada tabel berikut ini.

Tabel 2.1 Hasil kajian penelitian sebelumnya

Nomor Paten, Tahun	US20120203766 A1, 2012
Judul	<i>Faceted search result.</i>
Nama	John M. Hörnkvist, Wayne R. Loofbourrow, Eric R. Koebler
Gambaran umum penelitian	Sebuah paten mengenai sistem yang memberikan sejumlah hasil pencarian yang cocok dengan permintaan pencarian. Hasil kemudian dikategorikan menjadi beberapa ragam kategori serta sistem menampilkan subset pertama dari setiap keragaman kategori. Kemudian sistem mendapatkan input kedua dari pengguna yang memilih salah satu dari hasil keragaman kategori. Sistem kemudian

	merespon input kedua dan memproses data untuk ditampilkan, yaitu subset kedua dimana subset kedua ini adalah pengkategorian ke dalam satu atau lebih subkategori dari kategori yang telah dipilih [2].
Keterkaitan penelitian	Penelitian ini menggunakan pencarian dan penyaringan kategori yang saling berhubungan serta menghasilkan beberapa pilihan subkategori dari kategori yang telah disaring, sehingga didapatkan hasil yang sesuai dengan pencarian pengguna. Dengan mengacu pada penelitian ini, penulis dapat mempelajari dan memahami cara untuk melakukan <i>tagging</i> serta pengelompokan pencarian.

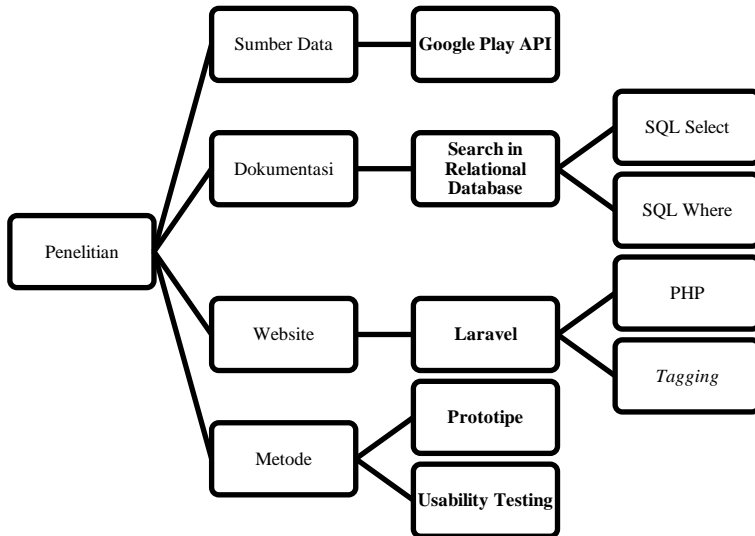
Tabel 2.2 Hasil kajian penelitian sebelumnya

Dipublikasikan pada, Tahun	<i>Lecture Notes in Business Information Processing</i> , 2010
Judul	<i>Faceted Wikipedia Search</i> .
Nama	Rasmus Hahn, Christian Bizer, Christopher Sahnwaldt, Christian Herta, Scott Robinson, Michaela Bürgle, Holger Düwiger, Ulrich Scheel
Gambaran umum penelitian	Dalam Wikipedia terdapat banyak artikel, disamping dalam bentuk teks, Wikipedia juga mengandung beberapa tipe informasi yang terstruktur, dimana yang paling bernilai dari informasi-informasi yang ada adalah Wikipedia infobox yang berisi atrikel-artikel yang paling relevan sebagai tabel atribut yang ditampilkan di pojok kanan atas laman Wikipedia. Data infobox tidak digunakan oleh mesin pencarian Wikipedia sehingga tidak bisa dicari,

	mesin pencari standar seperti Google dan Yahoo juga tidak dapat memanfaatkan data tersebut. Penelitian ini membentuk antarmuka alternatif untuk pencarian Wikipedia, dengan memanfaatkan pencarian teks, navigasi faset/ragam, pilihan penyaringan dari pengguna, serta hasil dari pencarian. Sehingga dengan memungkinkan pengguna untuk melakukan <i>query</i> pada Wikipedia seperti pada basis data yang terstruktur, mesin pencarian yang dibangun dapat mengeksploitasi intelegensia kolektif milik Wikipedia [6].
Keterkaitan penelitian	Penelitian ini menggunakan struktur yang diekstrak untuk menciptakan <i>query</i> pencarian serta keterlibatan pengguna dalam menentukan filter/penyaringan untuk menampilkan <i>faceted value</i> yang dipilih sehingga mendapatkan informasi yang sesuai dengan kebutuhan pencari. Dengan mengacu pada penelitian ini, penulis dapat mengimplementasikan metode yang digunakan untuk dapat merancang pencarian dan penyaringan konteks dalam penelitian penulis.

2.2. Dasar teori

Bagian ini membahas teori dan konsep yang berkaitan dengan penelitian tugas akhir. Relevansi dan dasar teori yang dibahas pada penelitian ini dapat dilihat pada Gambar 2.1.



Gambar 2.1. Bagan dasar teori

Dasar teori disusun untuk menjawab pertanyaan berikut:

1. Adakah cara cepat untuk mengambil data aplikasi dari Google Play?
2. Bagaimana mencari dan menyortir info yang berada pada basis data relasional?
3. Framework apakah yang dapat digunakan untuk mempercepat pengembangan aplikasi web dan membantu merapikan arsitektur?
4. Bagaimana menguji rancangan yang dikembangkan kepada pengguna?
5. Bagaimana menguji kemudahan penggunaan dari aplikasi yang dikembangkan kepada pengguna

2.2.1. Google play api

Google Play adalah layanan distribusi digital, yang juga merupakan toko media digital, Google Play (yang pada awalnya bernama Android Market), dikelola dan

dikembangkan oleh Google. Google Play berfungsi sebagai toko aplikasi resmi khusus untuk sistem operasi *mobile* android, yang memungkinkan pengguna untuk mencari dan mengunduh aplikasi yang dikembangkan dengan Android SDK dan dipublikasikan melalui Google. Google Play juga berfungsi sebagai toko media digital yang menawarkan musik, majalah, buku, film, dan program televisi [7].

Google Play API adalah *Application Programming Interface* yang disediakan untuk memudahkan pembangunan perangkat lunak yang berhubungan dengan data Google Play. Google Play API menyediakan metode untuk melakukan pengambilan data langsung dari Google Play yang akan digunakan dalam penelitian ini sebagai berikut:

- *List*: Mengambil daftar aplikasi dari suatu koleksi dalam Google Play.

```
GET: /list/:collection/:category/:age/:lang/:country/:num/:start
```

- *Suggest*: Mendapatkan 5 sugesti dari hasil pencarian kata kunci.

```
GET: /suggest/:term
```

- *Similar*: Mendapatkan data aplikasi terkait dari kata kunci yang diinput.

```
GET: /similar/:appId/:lang
```

- *Permissions*: Mendapatkan *permission* per aplikasi yang dispesifikan.

```
GET: /permission/:appId/:lang/:short
```

Hasil dari parameter dan metode ini adalah data Google Play yang ditampilkan dalam bentuk json [8]. Hasil dalam bentuk json ini adalah data yang diperlukan oleh penelitian untuk mengembangkan aplikasi pencarian dan penyaringan produk.

2.2.2. Search in relational database

Pencarian dalam basis data relasional menerapkan teknik pencarian kata kunci dalam bentuk basis data teks/*information retrieval* (IR) ke basis data relasional (DB). Teknik ini merupakan tugas yang menantang karena terdapat dua jenis basis data yang berbeda. Pertama, yaitu dalam basis data teks, unit informasi yang dicari oleh pengguna adalah dokumen. Untuk permintaan kata kunci tertentu, sistem IR akan menghitung skor numerik untuk setiap dokumen dan menentukan peringkat dokumen dengan skor ini. Dokumen-dokumen dengan peringkat teratas dikembalikan sebagai jawaban. Dalam basis data relasional, informasi disimpan dalam bentuk kolom, tabel, *primary key*, serta hubungan *foreign key*. Unit logis dari jawaban yang dibutuhkan oleh pengguna sifatnya tidak hanya terbatas pada nilai kolom individu atau bahkan baris individu saja, tetapi juga bisa beberapa baris dan kolom yang telah digabung.

Dengan menerapkan pencarian dalam basis data relasional, pengguna tidak perlu tahu skema basis data atau menggunakan bahasa *query* yang terstruktur untuk melakukan pencarian. Sebaliknya, pengguna hanya perlu menyerahkan daftar kata kunci dengan menggunakan antarmuka yang sangat sederhana, dan mesin pencari menampilkan hasil dokumen yang di-*ranking* berdasarkan relevansi dengan permintaan dari teks basis datanya. Dengan demikian pencarian data/dokumen yang tersimpan dalam basis data akan menjadi lebih mudah [9]. *Search in Relational Database* merupakan dokumentasi untuk mempelajari dan mengembangkan pencarian dan penyaringan aplikasi yang dibangun pada penelitian ini terutama pada bagian SQL untuk *query select* dan *where*.

2.2.3. Laravel

Laravel adalah sebuah *Framework* PHP 5.4 yang bersifat *opensource* dengan lisensi dibawah MIT *License*. Laravel dibuat untuk membantu para *developer* khususnya dalam membuat sebuah web dengan sintaks yang sederhana, elegan, ekspresif dan menyenangkan. Dengan Laravel, tugas-tugas umum *developer* dapat dikurangi pada sebagian besar proyek-proyek web seperti *routing*, *session* dan *caching* [10]. Laravel menyediakan kemungkinan pengembangan yang cepat dan aman. Kode yang ditulis dengan Laravel dapat digunakan lagi, dan lebih terorganisir. Terdapat pula banyak produk dan paket untuk mendukung pengembangan dengan Laravel [11]. Pada penelitian ini, tujuan utama dari Laravel adalah mempermudah *coding* dalam mengembangkan aplikasi web berbasis PHP/HTML serta memberikan label pada penyaringan produk yang dibangun. Laravel juga membantu untuk menyusun aplikasi yang dibangun dengan lebih terstruktur.

2.2.4. Prototyping

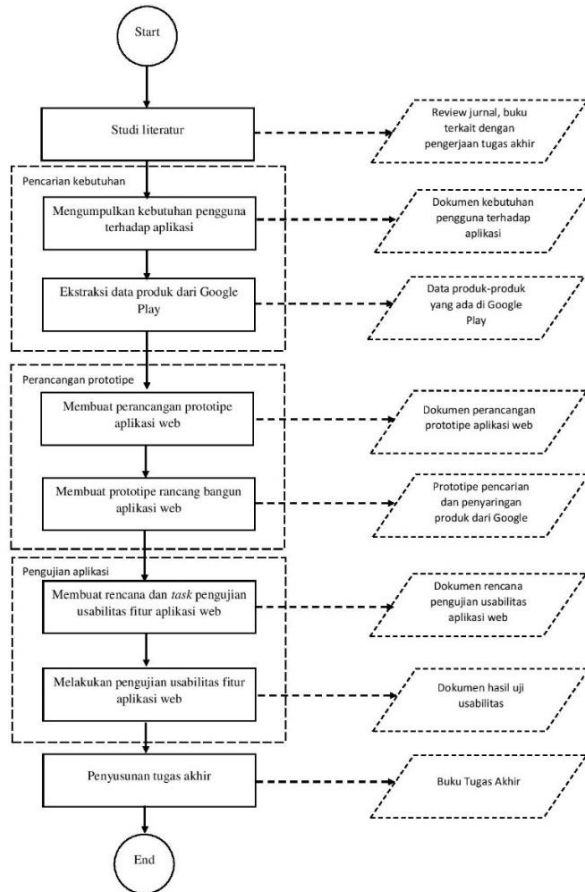
Prototyping merupakan strategi yang sangat baik untuk melakukan desain yang efektif dengan menciptakan desain produk. Prototipe bukan sebuah produk yang sudah jadi secara utuh, melainkan produk yang dapat digunakan fungsifungsinya sebagai uji coba agar memudahkan pendesain dalam mendapatkan *feedback*. Harus diperhatikan dengan baik bahwa tujuan terpenting dari prototipe adalah mendapatkan *feedback*. Dengan melakukan prototipe, pendesain dapat terus menyempurnakan produk akhir yang ingin dihasilkan berdasarkan *feedback* yang diterima dari partisipan [4]. Prototipe tidak dimaksudkan sebagai versi sempurna dari produk yang dirancang. Seringkali tidak mungkin menggunakan bahan yang sama yang akan dimasukkan ke dalam produk yang sebenarnya, dan mungkin tidak ada proses pabrik untuk membangunnya [12].

2.2.5. Usability testing

Setelah prototipe dihasilkan dilakukan pengujian usability. Dengan pengujian usability, produk yang telah dihasilkan dapat dievaluasi. Proses pengujian usability ini dilakukan oleh partisipan, dengan cara partisipan mencoba menyelesaikan beberapa *task* sambil diawasi oleh pengamat. Tujuannya adalah menemukan masalah usability dari produk yang dihasilkan dan mengumpulkan data kuantitatif serta kualitatif yang ditentukan dari kepuasan/harapan partisipan terhadap produk yang diujikan [13]. Dalam melakukan pengujian untuk prototipe terdapat cara yang mudah, yaitu dengan menggunakan *System Usability Testing* (SUS). SUS adalah *tools* untuk mengukur usability system dengan kuesioner terdiri dari 10 pertanyaan dengan 5 pilihan jawaban. Walau jumlah responden sangat sedikit (jumlahnya hanya 2 responden saja) SUS sudah terbukti *reliable* dan *valid*. Penilaian SUS dilakukan dengan cara; untuk pertanyaan nomor ganjil skor pilihan responden dikurangi 1 (misal responden milih **setuju** yang memiliki skor 4, maka $4 - 1$), dan untuk pertanyaan nomor genap cara menghitungnya adalah 5 dikurangi skor pilihan responden (misal responden milih **tidak setuju** yang memiliki skor 2, maka $5 - 2$). Kemudian skor dari pertanyaan nomor 1 sampai 10 ditotal dan dikali 2,5 [14].

BAB III METODOLOGI

Bagian ini akan menjelaskan metodologi dan alur pengerjaan tugas akhir. Gambaran metodologi ini akan berguna sebagai pedoman selama pengerjaan tugas akhir yang dapat dilihat pada Gambar 3.1 berikut :



Gambar 3.1 Metode penelitian

3.1. Tahapan

Penjelasan setiap tahapan dari metodologi adalah sebagai berikut:

3.1.1. Studi literatur

Pada tahap studi literatur, penulis mengumpulkan informasi yang nantinya digunakan untuk mengusulkan solusi terkait dengan permasalahan pencarian dan penyaringan kebutuhan pengguna. Pengumpulan data dan informasi sendiri dilakukan dengan cara membaca referensi dari buku dan penelitian-penelitian yang telah dilakukan sebelumnya. Tujuan dari tahap ini agar penulis dapat memahami dasar teori yang berhubungan dengan permasalahan dan dapat mempermudah dalam menemukan solusi yang tepat.

3.1.2. Pencarian kebutuhan

Tahapan ini dilakukan untuk mengetahui kebutuhan dari pengguna dalam menemukan produk yang sesuai dengan kebutuhan dirinya. Setelah mendapatkan data kebutuhan pengguna dan mempelajari kajian mengenai teknik pembangunan pencarian dan penyaringan produk maka diperlukan ekstraksi data kategori, *app permission*, *similar app*, serta *sugessted* pada Google Play.

3.1.3. Perancangan prototipe

Setelah mendapatkan data ekstraksi produk dari Google Play, maka langkah selanjutnya adalah membuat perancangan prototipe dan merancang prototipe sesuai dengan kebutuhan pengguna mengenai pencarian dan penyaringan produk. Tujuan utama dari Tugas Akhir ini adalah pembuatan aplikasi web berbasis PHP/HTML yang dapat membantu pencarian dan penyaringan produk. Luaran dari tahapan perancangan prototipe adalah web aplikasi berbasis PHP/HTML yang mampu menampilkan pencarian dan penyaringan produk serta *tagging* kategori yang sesuai dengan hasil pencarian produk.

3.1.4. Pengujian aplikasi

Selanjutnya aplikasi akan diuji usabilitasnya. Apabila hasil uji usability lebih rendah dari yang diharapkan maka desain akan diubah sesuai dengan harapan pengguna. Namun, dalam konteks tugas akhir ini, hasil pengujian usability akan digunakan sebagai bahan evaluasi perbaikan sistem di masa yang akan datang.

3.1.5. Hasil dan pembahasan

Setelah prototipe dibuat, maka tahapan selanjutnya adalah melakukan uji validasi terhadap prototipe yang digunakan dengan metode *usability testing*, dengan melakukan survei terhadap calon pengguna dari fitur ini. Jika prototipe telah dapat berjalan dengan baik dan survei yang dilakukan mendapatkan hasil yang telah diinginkan, maka dapat diambil kesimpulan terhadap penelitian ini.

Tahapan selanjutnya adalah penulisan Buku Tugas Akhir sebagai dokumentasi dari penelitian ini. penulisan buku dilakukan bersamaan dengan seluruh tahapan penelitian yang dilakukan. Luaran yang dihasilkan adalah Buku Tugas Akhir yang diharapkan dapat bermanfaat bagi berbagai pihak dan menjadi referensi untuk penelitian yang berkaitan dengan bidang yang sama selanjutnya.

Halaman ini sengaja dikosongkan

BAB IV

ANALISIS DAN PERANCANGAN

Bagian ini akan memberikan gambaran mengenai perancangan terhadap sistem yang akan dibangun sebelum melakukan tahapan implementasi pada penelitian tugas akhir. Perancangan pada bab ini meliputi subyek dan obyek penelitian, pemilihan subyek dan obyek penelitian, dan bagaimana tahapan penelitian akan dilakukan agar tetap sesuai dengan alur pengerjaan ketika akan melakukan tahapan implementasi.

4.1. Pencarian kebutuhan

Pada bagian ini akan dijelaskan mengenai pencarian kebutuhan yang bertujuan sebagai dasar perancangan dan pengembangan aplikasi pada penelitian ini.

4.1.1 Wawancara kebutuhan

Wawancara digunakan untuk mencari kebutuhan pengguna perangkat serta ekspektasi pengguna terhadap aplikasi yang akan dibangun. Untuk mencari kebutuhan pengguna perangkat serta ekspektasi pengguna terhadap aplikasi diberikan wawancara mengenai kebutuhan terhadap aplikasi pencari produk pada Google Play yang dapat menambahkan dan mengurangi penyaringan agar pencarian semakin tepat. Responden wawancara adalah pemilik perangkat telpon genggam berbasis android atau pernah menggunakan aplikasi Google Play. Dari wawancara kebutuhan dilakukan analisis terhadap kebutuhan pengguna dari aplikasi yang akan dikembangkan. Wawancara dilakukan pada tanggal 21 Maret 2017 kepada Ramadhan. Ramadhan adalah seorang pengguna android. Beliau memiliki masalah ketika mencari produk aplikasi di Google Play, ketika mencari produk aplikasi Google Play hanya menyediakan kolom masukan kata kunci saja. Karena hanya terdapat pencarian dengan menggunakan kolom kata kunci, Ramadhan sering kebingungan memasukan

kata kunci dan berharap sudah ada contoh pencarian yang bisa dipilih sehingga mungkin produk yang dicari sama dengan yang Ramadhan inginkan. Selain itu Ramadhan merasa perlu mempersempit pencarian agar tidak terlalu banyak produk aplikasi yang muncul pada saat mencari.

Tabel 4.1 Kebutuhan pengguna

ID user needs	Kebutuhan pengguna
Tujuan: Membuat aplikasi yang dapat menyaring pencarian kata kunci.	
UN001	Pengguna aplikasi dapat mencari produk aplikasi dengan kata kunci.
UN002	Pengguna aplikasi dapat menambahkan penyaringan dari kata kunci pencarian.
UN003	Pengguna aplikasi dapat dengan mudah menemukan produk yang sesuai dengan kebutuhannya
Tujuan: Menyediakan sistem yang dapat memepermudah dan mempercepat pencarian	
UN004	Pengguna aplikasi dapat menyimpan pencarian kata kunci.
UN005	Pengguna aplikasi dapat mencari produk aplikasi dengan pencarian yang sudah tersipan.

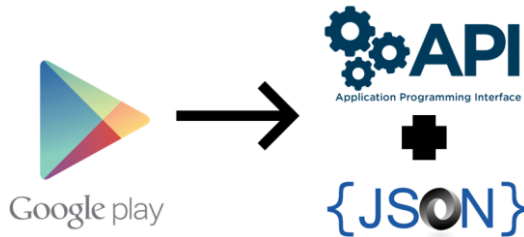
4.2. Perancangan pengambilan data

Pada bagian ini akan dijelaskan mengenai perancangan pengambilan data yang bertujuan sebagai panduan ketika akan melakukan implementasi pada penelitian ini.

4.2.1. Data yang akan diambil

Pengambilan data dilakukan untuk mendapatkan produk aplikasi yang akan digunakan dalam pencarian dan penyaringan. Untuk melakukan proses pengambilan data, dapat digunakan API yang diciptakan sesuai dengan stuktur

situs Google Play sehingga data dapat dengan mudah diambil. API yang digunakan berasal dari `gooplayapi.herokuapp.com` yang diciptakan oleh Natanael Yabes W [8]. Data yang diambil adalah seluruh struktur data setiap produk aplikasi yang ada pada Google Play. Alur pengambilan data yang dilakukan pada Google Play API dapat dilihat pada Gambar 4.1:



Gambar 4.1 Alur pengambilan data sebelum dimodifikasi dari google play

Data yang dibutuhkan oleh penelitian ini berupa data-data produk aplikasi. Pengambilan data pada Google Play API dilakukan secara *real time* dengan stuktur seperti tabel dibawah:

Tabel 4.2 Deskripsi data atribut tabel applications

Nama Tabel	Nama kolom	Tipe data	Keterangan
applications	id	Int(11)	Primary Key untuk tabel applications
applications	title	Varchar (100)	Nama dari aplikasi
applications	summary	Varchar (100)	Outline/abstrak/tag dari aplikasi
applications	icon	Varchar (100)	Gambar/logo/symbol dari aplikasi

applications	price	Varchar (100)	Harga jual dari aplikasi
applications	free	Tinyint (1)	Aplikasi tidak memiliki harga jual (true/false)
applications	minInstalls	Int(11)	Instalasi minimal dari aplikasi
applications	maxInstalls	Int(11)	Instalasi maksimal dari aplikasi
applications	score	Decimal (10,2)	Rata-rata skor rating dari aplikasi
applications	reviews	Int(11)	Jumlah review yang diberikan untuk aplikasi
applications	developer	Varchar (100)	Nama dari pengembang aplikasi
applications	developerEmail	Varchar (100)	Alamat email milik pengembang aplikasi
applications	developerWebsite	Varchar (100)	Alamat situs milik pengembang aplikasi
applications	update	Varchar (100)	Tanggal terakhir aplikasi diperbarui
applications	genre	Varchar (100)	Kategori dari aplikasi
applications	genreId	Varchar (100)	Id pada googleplay untuk

			kategori aplikasi
applications	description	Text	Deskripsi/gambaran menyeluruh dari aplikasi
applications	descriptionHTML	Text	Deskripsi/gambaran menyeluruh dari aplikasi versi HTML
applications	offersIAP	Tinyint (1)	Aplikasi menyediakan pembelian di dalam aplikasi (In App Purchase) (true/false)
applications	adSupported	Tinyint (1)	Terdapat iklan dalam aplikasi
applications	androidVersionText	Varchar (100)	Versi android yang dibutuhkan agar aplikasi dapat diinstall dalam bentuk text pada googleplay
applications	androidVersion	Varchar (100)	Versi android yang dibutuhkan agar aplikasi dapat diinstall
applications	contentRating	Varchar (100)	Rating usia untuk pengguna dari aplikasi
applications	video	Varchar (100)	Link video dari aplikasi
applications	recentChanges	Varchar (100)	Gambaran perubahan yang terjadi pada versi

			aplikasi saat ini
applications	preregister	Tinyint (1)	Aplikasi dapat dicoba terlebih dahulu (true/false)
applications	url	Varchar (100)	Link aplikasi dapat diunduh pada googleplay
applications	appId	Varchar (100)	Id aplikasi yang diberikan oleh googleplay untuk aplikasi

Tabel 4.3 Deskripsi data atribut tabel applications_facet

Nama Tabel	Nama kolom	Tipe data	Keterangan
applications_facets	id	Int(11)	Primary Key untuk tabel applications_facets
applications_facets	name	Varchar(30)	Nama facets untuk filter

Tabel 4.4 Deskripsi data atribut tabel applications_fact

Nama Tabel	Nama kolom	Tipe data	Keterangan
applications_fact	app_id	Int(11)	Primary Key untuk tabel applications_fact
applications_fact	facet_id	Int(11)	Foreign Key untuk tabel applications_fact ke

			tabel applications_facets
applications_fact	facet_name	Varchar(30)	Nama facet yang di dapat dari tabel applications_facets
applications_fact	value	Varchar(30)	Nilai facet yang digunakan untuk filter

Teknik yang digunakan untuk pengambilan data dari Google Play API ini menggunakan Bahasa pemrograman PHP dan JSON yang kemudian disimpan kedalam basis data MySQL. Dari proses pengambilan data tersebut, sebanyak 1074 data berhasil dikumpulkan. Atribut pada Tabel 4.1 menampung data produk aplikasi, sedangkan atribut pada Tabel 4.2 dan 4.3 menampung data untuk penyaringan produk aplikasi yang semuanya disimpan kedalam basis data MySQL.

Data yang dimasukkan ke basis data ini bersifat umum dan spesifik. Data umum ini adalah data semua produk aplikasi pada Google Play yang hanya menampilkan data secara umum mengenai aplikasi tersebut. Data spesifik adalah data yang berisikan satu aplikasi namun menampilkan data secara detail. Data umum dan spesifik kemudia dijadikan 1 tabel, yaitu tabel applications pada basis data dengan atribut yang dapat dilihat pada Tabel 4.1. Proses pengambilan data dari API + JSON memiliki alur yang dapat dilihat pada Gambar 4.2:



Gambar 4.2 Alur pengambilan data dari api + json

Semua data yang disimpan dapat digunakan untuk merancang prototipe aplikasi pencarian dan penyaringan.

4.2.2. Desain basis data

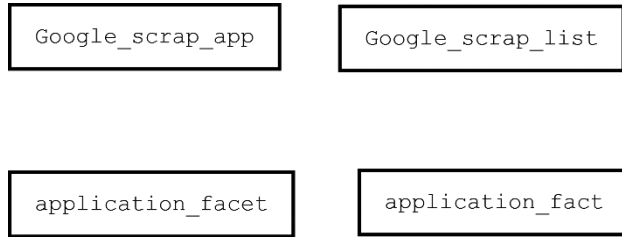
Pada bagian ini akan dijelaskan mengenai desain basis data yang memiliki tujuan merancang basis data yang terstruktur dengan membuat relasi antar tabel yang pada pengerjaannya memiliki berbagai tahapan agar kolom-kolom dalam tabel tersebut hanya berisikan data-data yang tidak redundan.

4.2.2.1. Perancangan basis data konseptual

Perancangan basis data konseptual adalah proses membuat suatu model informasi yang digunakan pada suatu perusahaan, yang terlepas dari semua pertimbangan fisik. Langkah-langkah di dalam perancangan basis data konseptual, yaitu membangun model data konseptual untuk setiap user view [15].

a. Identifikasi tipe entiti

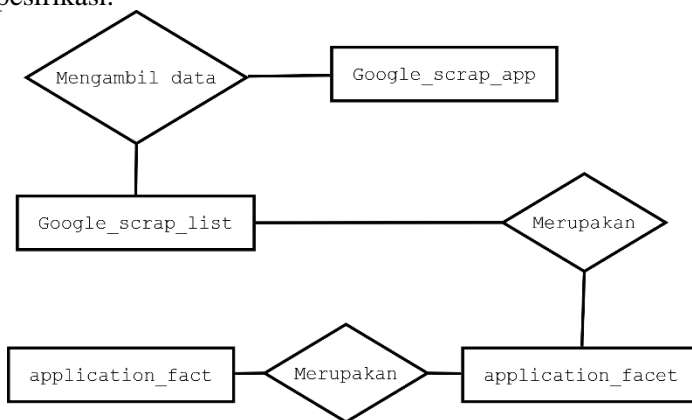
Tahapan ini bertujuan untuk mengidentifikasi tipe utama yang diperlukan dari sudut pandang tertentu. Salah satu cara yang dapat digunakan adalah dengan mengidentifikasi kata benda utama dan objek-objek yang berkaitan dengan spesifikasi seperti gambar berikut:



Gambar 4.3 Entiti yang saling berkaitan

b. Indentifikasi tipe relasi

Tahapan ini bertujuan untuk mengidentifikasi relasi penting yang ada diantara berbagai tipe entiti yang telah diidentifikasi. Salah satu metode yang digunakan adalah dengan mempelajari kata kerja yang berkaitan dengan entiti, tipe relasi, dan spesifikasi.

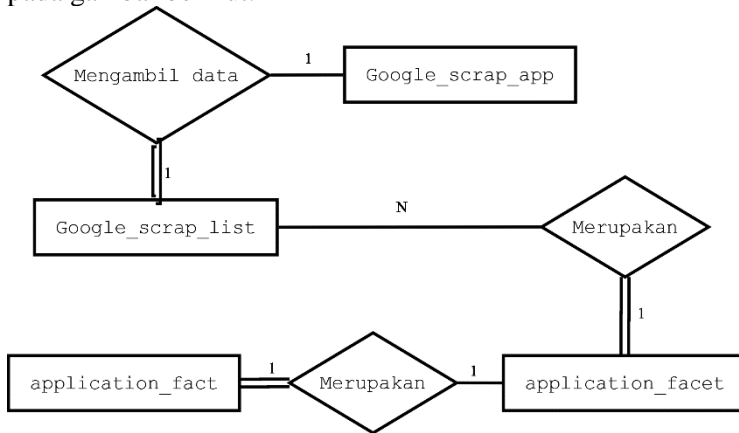


Gambar 4.4 Identifikasi relasi antara entiti

Gambar 4.4 merupakan identifikasi relasi antara tiap entiti yang telah digambarkan pada Gambar 4.3.

c. Indentifikasi dan menghubungkan atribut dengan entity ataupun dengan tipe relasi

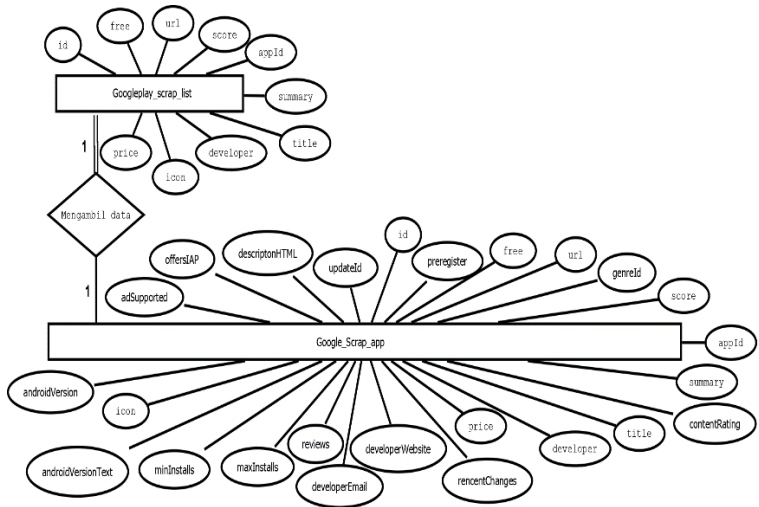
Tahapan ini bertujuan untuk menghubungkan atribut dengan tipe entiti dan relasi yang sesuai. Dalam menentukan atribut harus mampu mengidentifikasi atribut *simple* atau *composite*, atribut *single* atau *multi-valued*, dan atribut turunan seperti pada gambar berikut:



Gambar 4.5 Menghubungkan atribut entiti

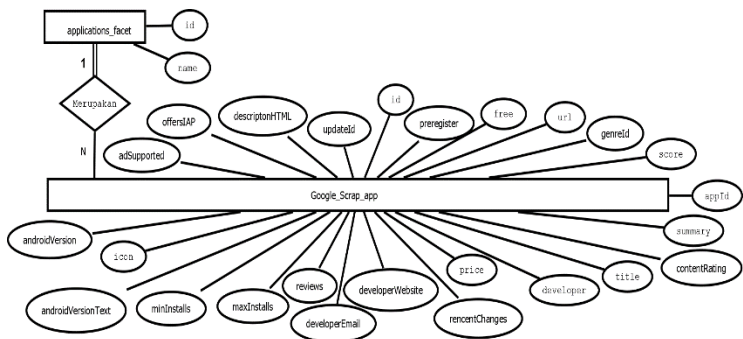
d. Menentukan domain atribut

Tahapan ini bertujuan untuk menentukan wewenang bagi atribut dalam model data konseptual. Domain adalah satu kumpulan nilai yang menggambarkan nilai-nilai dari satu/lebih atribut.



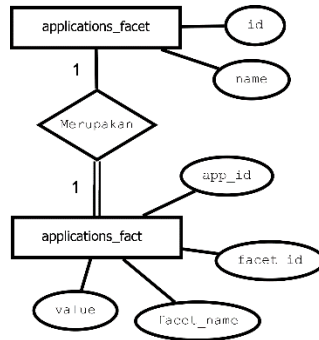
Gambar 4.6 Domain atribut Googleplay_scrap_list dan Googleplay_scrap_app

Gambar 4.6 merupakan domain atribut yang ada pada Googleplay_scrap_app dan Googleplay_scrap_list beserta hubungannya.



Gambar 4.7 Domain atribut application_facet dan Google_scrap_app

Gambar 4.7 merupakan domain atribut yang ada pada Googleplay_scrap_app dan applications_facet beserta hubungannya.



Gambar 4.8 Domain atribut applications_facet dan applications_fact

Gambar 4.8 merupakan domain atribut yang ada pada applications_facets dan applications_fact beserta hubungannya.

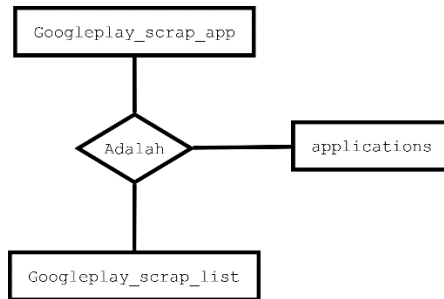
e. Menentukan atribut *candidate key* dan *primary key*

Tahapan ini bertujuan untuk mengidentifikasi *candidate key* untuk setiap tipe relasi. Jika terdapat lebih dari satu *candidate key*, maka dapat dipilih salah satu atribut untuk menjadi *primary key*. Untuk Google Play memiliki *primary key*, yaitu id dan *candidate key* yaitu appId. Pada tabel applications_facets memiliki *primary key*, yaitu app_id. Pada tabel applications_facet data tidak terdapat *primary key* maupun *candidate key*.

f. Mempertimbangkan pengguna konsep pemodelan yang lebih baik

Tahapan ini bertujuan untuk mempertimbangkan konsep pemodelan yang lebih baik seperti spesialisasi, generalisasi, agregasi, atau komposisi. Pada tahap ini yang terjadi hanyalah proses generalisasi, dimana terjadi proses peleburan tabel

googleplay_scrap_app dan googleplay_scrap_list menjadi tabel applications. Ini dikarenakan atribut yang ada di googleplay_scrap_list memiliki kesamaan terhadap googleplay_scrap_app. Demi menghemat memori dan menghilangkan redundansi, maka dibutuhkan proses generalisasi seperti pada Gambar 4.9:



Gambar 4.9 melakukan generalisasi menjadi tabel applications

g. Memeriksa dengan model redundansi

Tahapan ini bertujuan untuk memeriksa keberadaan redundansi dalam model basis data. Aktivitas yang dilakukan pada tahapan ini adalah menentukan ulang relasi 1:1 dan membuang relasi yang redundan. Pada tahap ini tidak ada perubahan pada basis data konseptual dikarenakan kondisi tabel yang ada tidak memerlukan perlakuan dua aktivitas.

h. Data dictionary

Data dictionary atau *system catalog*, bisa diartikan sebagai hasil kompilasi *data definition language* yang kumpulan tabelnya tersimpan secara kolektif.

Biasanya, di dalam sebuah *data dictionary* terdapat beberapa hal berikut:

- Definisi dari semua objek skema yang terdapat di dalam sebuah basis data (tabel, *view*, *index*, *cluster*, sinonim, urutan, prosedur, dll).

- Berapa banyak alokasi data yang digunakan oleh objek skema tersebut.
- Nilai default untuk setiap kolom.
- Nama dari pengguna.
- Keuntungan dan peran dari setiap pengguna yang telah diperbolehkan.
- Informasi yang telah diaudit, seperti siapa yang dapat mengakses atau memperbaharui objek skema tersebut.
- Dan informasi umum mengenai basis data lainnya.

Data dictionary mempunyai tiga pengguna utama, yaitu:

- Administrator yang dapat mengakses *data dictionary* tersebut untuk mengetahui informasi mengenai pengguna, objek skema dan struktur penyimpanan dari basis data.
- Administrator yang dapat memodifikasi *data dictionary* setiap kali ada DDL (*Data Definition Language*) yang ditambahkan.
- Pengguna lainnya yang dapat menggunakan *data dictionary* sebagai referensi yang dapat dibaca untuk mengetahui informasi apapun mengenai basis data.

Pembuatan *data dictionary* berguna untuk mengetahui jenis tipe apa yang akan digunakan pada setiap kolom data pada basis data yang digunakan. *Data dictionary* merupakan penjelasan lebih rinci ERD (*Entity Relation Diagram*) dari sebuah basis data. Pada penelitian kali ini, akan ada tiga tabel yang akan digunakan, yaitu tabel *applications* untuk menyimpan data produk aplikasi dari Google Play, tabel *applications_facet* untuk menyimpan data acuan penyaringan, dan tabel *applications_fact* untuk menyimpan data produk aplikasi yang sudah diberi acuan penyaringan. *Data dictionary* pada tabel *applications* dapat dilihat pada tabel berikut:

Tabel 4.5 Data dictionary tabel applications

Table	Column	Xtype	Data Type	Len	Null	Description
applications	id	PK, FK	Int	11	N	Primary Key untuk tabel applications
applications	title		Varchar	100	N	Nama dari aplikasi
applications	summary		Varchar	100	N	Outline/abstrak/tag dari aplikasi
applications	icon		Varchar	500	N	Gambar/logo /simbol dari aplikasi
applications	price		Tinyint	1	N	Harga jual dari aplikasi
applications	minInstalls		Int	11	N	Aplikasi tidak memiliki harga jual (true/false)
applications	maxInstalls		Int	11	N	Instalasi minimal dari aplikasi
applications	score		Decimal	10, 2	N	Instalasi maksimal dari aplikasi
applications	reviews		Int	11	N	Rata-rata skor rating dari aplikasi
applications	developer		Varchar	42	N	Jumlah review yang diberikan untuk aplikasi

applications	developerEmail		Varchar	45	N	Nama dari pengembang aplikasi
applications	developerWebsite		Varchar	45	N	Alamat email milik pengembang aplikasi
applications	update		Varchar	45	N	Alamat situs milik pengembang aplikasi
applications	genre		Varchar	45	N	Tanggal terakhir aplikasi diperbarui
applications	genreID		Varchar	45	N	Kategori dari aplikasi
applications	description		Text		N	Id pada googleplay untuk kategori aplikasi
applications	descriptionHTML		Text		N	Deskripsi/gambaran menyeluruh dari aplikasi
applications	offersIAP		Tinyint	1	N	Deskripsi/gambaran menyeluruh dari aplikasi versi HTML
applications	adSupported		Tinyint	1	N	Aplikasi menyediakan pembelian di dalam aplikasi (In

						App Purchase) (true/false)
applications	androidVersionText		Varchar	45	N	Terdapat iklan dalam aplikasi
applications	androidVersion		Varchar	10	N	Versi android yang dibutuhkan Agar aplikasi dapat diinstall dalam bentuk text pada googleplay
applications	contentRating		Varchar	10	N	Versi android yang dibutuhkan agar aplikasi dapat diinstall
applications	video		Varchar	100	N	Rating usia untuk pengguna dari aplikasi
applications	recentChanges		Varchar	45	N	Link video dari aplikasi
applications	preregister		Tinyint	1	N	Gambaran perubahan yang terjadi pada versi aplikasi saat ini
applications	url		Varchar	100	N	Aplikasi dapat dicoba terlebih

						dahulu (true/false)
applic ations	appID		Varc har	100	N	Link aplikasi dapat diunduh pada googleplay

Data dictionary pada tabel applications_facets dapat dilihat pada tabel berikut:

Tabel 4.6 Data dictionary tabel applications_facets

Table	Colum n	Xty pe	Data Type	Len	Null	Description
applic ations _facets	id	PK	Int	11	N	Primary Key untuk tabel applications_ facets
applic ations _facets	name		Varc har	30	N	Nama facets untuk filter

Data dictionary pada tabel applications_facts dapat dilihat pada tabel berikut:

Tabel 4.7 Data dictionary tabel applications_fact

Table	Colum n	Xty pe	Data Type	Len	Null	Description
applic ations _fact	app_id		Int	11		Primary Key untuk tabel applications_ fact
applic ations	facet_i d		Int	11		Foreign Key untuk table

fact						applications fact ke tabel applications_ facets
applic ations _fact	facet_ name		Varc har	30		Nama facet yang di dapat dari tabel applications_ facets
applic ations _fact	value		Varc har	30		Nilai facet yang digunakan untuk filter

4.2.2.2. Perancangan basis data logikal

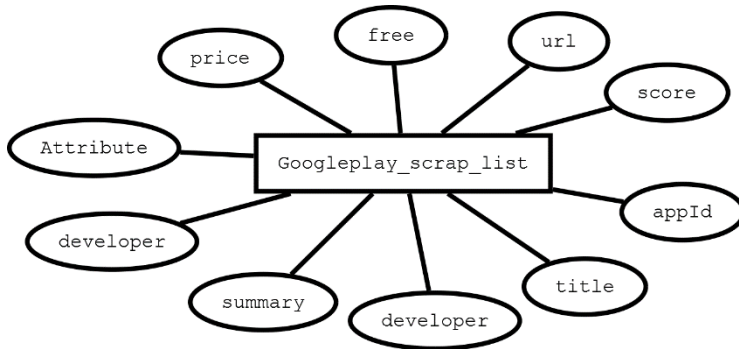
Perancangan basis data logikal adalah proses membangun suatu model informasi yang digunakan didalam perusahaan berdasarkan model data tersebut, tanpa memperhatikan pertimbangan penggunaan suatu DBMS dan perangkat keras lainnya. Tujuan dari perancangan ini adalah membangun model data logikal yang mengacu pada model data logikal dan juga memvalidasi model agar mendukung transaksi yang dibutuhkan oleh user [15].

i. Menentukan relasi untuk model data logikal

Tahapan ini bertujuan dalam membuat suatu relasi untuk model data logika yang menggambarkan entity, relasi, dan atribut yang telah di identifikasi. Turunan dari tahapan ini terbagi menjadi 9 proses dalam menentukan relasi model data.

Mapping of strong entity types

Dari proses *Mapping of strong entity types* dihasilkan Entitas Googleplay_scrap_list yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.10 merupakan Entitas Googleplay_scrap_list:



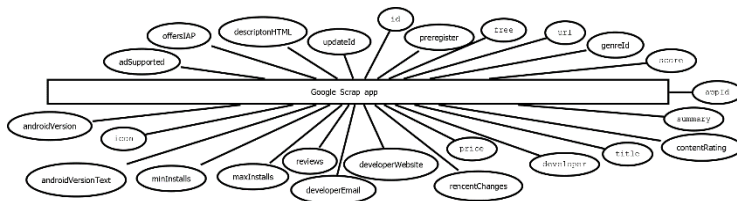
Gambar 4.10 Entitas dan atribut Google_scrap_app

Entitas `Googleplay_scrap_list` yang diubah kedalam MRD menjadi seperti Gambar 4.11:

appId	title	summary	icon	price	free	score	developer	url
-------	-------	---------	------	-------	------	-------	-----------	-----

Gambar 4.11 Bentuk MRD dari Googleplay_scrap_list

Dari proses *Mapping of strong entity types* dihasilkan Entitas `Google_scrap_app` yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.12 merupakan Entitas `Googleplay_scrap_app`:



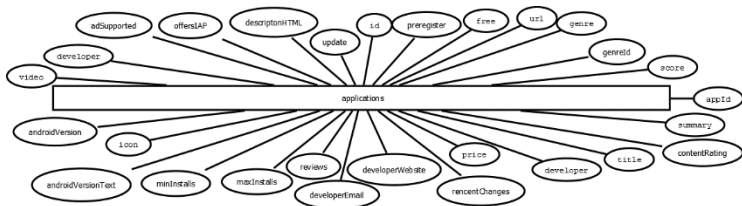
Gambar 4.12 Entitas dan atribut Googleplay_scrap_app

Entitas `Googleplay_scrap_list` yang diubah kedalam MRD menjadi seperti Gambar 4.13:

appId	description	descriptionHTML	minInstalls	maxInstalls
genre	genreId	developerEmail	developerWebsite	adSupported
androidVersion	androidVersionText	contentRating	reviews	
updated	preRegister	offersIAP	recentChanges	

Gambar 4.13 Bentuk MRD dari Googleplay_scrap_app

Dari proses *Mapping of strong entity types* dihasilkan Entitas applications yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.14 merupakan Entitas applications:



Gambar 4.14 Entitas dan atribut applications

Entitas applications yang diubah kedalam MRD menjadi seperti Gambar 4.15:

appId	description	descriptionHTML	minInstalls	maxInstalls
-------	-------------	-----------------	-------------	-------------

genre	genreId	developerEmail	developerWebsite	adSupported
-------	---------	----------------	------------------	-------------

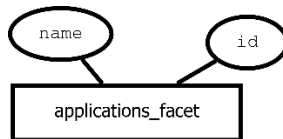
androidVersion	androidVersionText	contentRating	reviews
----------------	--------------------	---------------	---------

updated	preRegister	offersIAP	recentChanges	price	video
---------	-------------	-----------	---------------	-------	-------

id	title	summary	icon	score	developer	update	url
----	-------	---------	------	-------	-----------	--------	-----

Gambar 4.15 Bentuk MRD dari applications

Dari proses *Mapping of strong entity types* dihasilkan Entitas applications_facet yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.16 merupakan Entitas applications_facet:



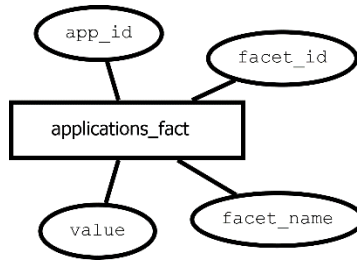
Gambar 4.16 Entitas dan atribut applications_facet

Entitas applications_facet yang diubah kedalam MRD menjadi seperti Gambar 4.17:

id	name
----	------

Gambar 4.17 Bentuk MRD dari applications_facets

Dari proses *Mapping of strong entity types* dihasilkan Entitas applications_fact yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.18 merupakan Entitas applications_fact:



Gambar 4.18 Entitas dan atribut yang dimiliki

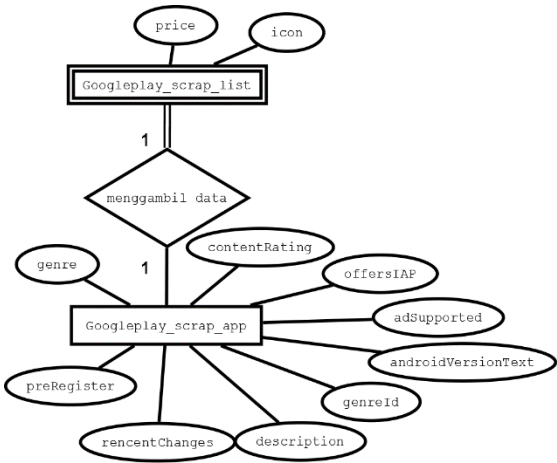
Entitas `applications_fact` yang diubah kedalam MRD menjadi seperti Gambar 4.19:

app_id	facet_id	facet_name	value
--------	----------	------------	-------

Gambar 4.19 Bentuk MRD dari `applications_fact`

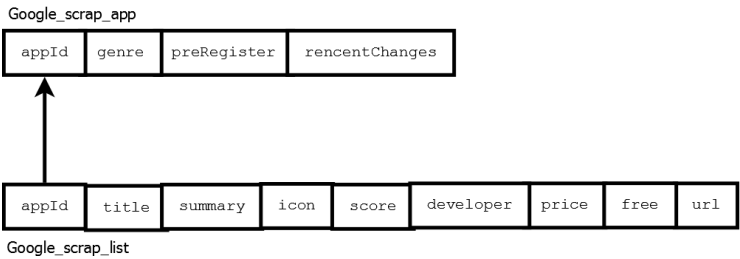
Mapping of weak entity types

Dari proses *Mapping of weak entity types* dihasilkan Entitas `googleplay_scrap_list` yang kemudian akan diubah kedalam bentuk MRD. Gambar 4.20 merupakan entitas `googleplay_scrap_list`:



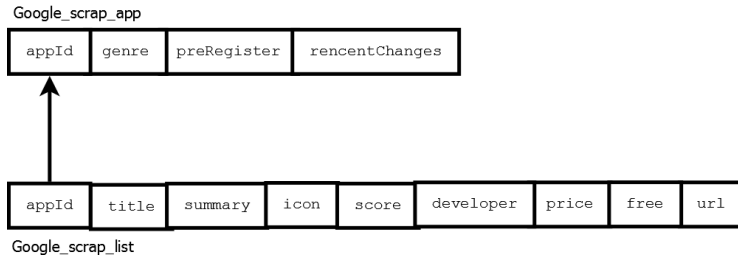
Gambar 4.20 Weak entity Googleplay_scrap_list dengan Googleplay_scrap_app

Weak Entity dari Googleplay_scrap_list yang berelasi dengan entitas Googleplay_scrap_app dalam MRD adalah Gambar 4.21:



Gambar 4.21 Relasi Googleplay_scrap_list dengan Googleplay_scrap_app dalam bentuk MRD

Entitas googleplay_scrap_list yang berelasi one to one dengan entitas googleplay_scrap_app dalam MRD adalah sebagai berikut:



Gambar 4.22 Relasi one-to-one Googleplay_scrap_list dengan Googleplay_scrap_app

One to many (1:N) tipe relasi binary

Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi binary 1:N.

One to one (1:1) tipe relasi rekursif

Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi rekursi 1:1.

***Superclass* atau *subclass* tipe relasi**

Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi *superclass* atau *subclass*.

***Many-to-many* (*:*) tipe relasi binary**

Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi binary M:N.

Tipe relasi kompleks

Tidak dilakukan karena dalam ERD tidak ada atribut yang memiliki relasi kompleks.

Atribut *multivalued*

Tidak dilakukan karena dalam ERD tidak ada atribut yang *multivalued*.

4.2.2.3. Perancangan basis data fisikal

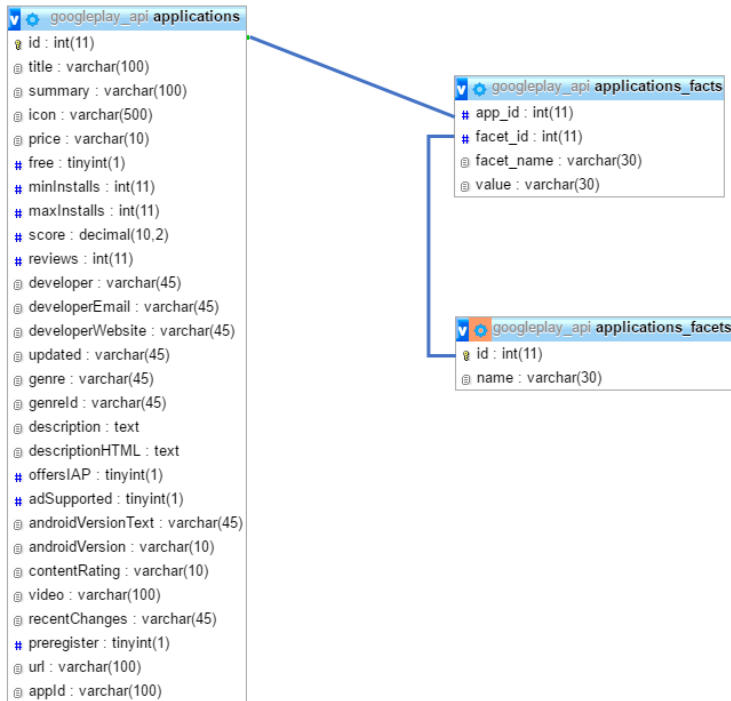
Perancangan basis data fisikal adalah proses pembuatan deskripsi implementasi dari basis data pada *media secondary storage*. perancangan ini menggambarkan relasi dasar organisasi file pengindeksan yang digunakan untuk memperoleh akses yang efisien pada data, serta batasan integritas dan ukuran yang berhubungan [15]. Langkah-langkah dalam metodologi perancangan basis data fisikal adalah sebagai berikut :

j. Menerjemahkan model data logikal ke target DBMS

Tahapan ini bertujuan untuk menghasilkan skema basis data relasional dari model data logikal yang dapat diimplementasikan kedalam suatu DBMS. Aktifitas yang dilakukan meliputi merancang relasi dasar, merancang representasi dari data turunan, dan merancang *general constraints*.

k. Merancang relasi dasar

Pada tahapan ini, dalam mentranslasi model data logikal pada bagian relasi, perlu penyesuaian pada atribut dan relasinya. Dalam hal ini, atribut yang ada bisa dilihat di data dictionary, lalu untuk relasi bisa dilihat pada Gambar 4.23:



Gambar 4.23 Relasi pada model data fisik

Relasi yang ada yaitu app_id dimana primary key berada di id tabel applications dan pada facet_id dimana primary key berada di id tabel applications_facet.

1. Merancang representasi dari data turunan

Tahapan ini tidak dilakukan karena pada data tidak terjadi data turunan.

4.2.3. Rancang bangun prototipe

Sesuai dengan tahapan ICONIX, hal pertama yang akan dibuat adalah antar muka pengguna, kemudian diturunkan menjadi *use case diagram*. Selanjutnya *class diagram* dibuat dari hasil penurunan *use case diagram* dari antar muka pengguna yang

telah dibuat. Kebutuhan dari perancangan prototipe didapatkan dari proses pencarian kebutuhan.

4.2.4. Antar muka pengguna

Berikut ini merupakan desain antar muka pengguna aplikasi. Antar muka yang dibuat terdiri dari dua antar muka; halaman pencarian dan halaman hasil pencarian. Untuk setiap antar muka pengguna yang dibuat, akan dijadikan acuan dalam pengembangan *use case*.

4.2.4.1. Halaman pencarian

Google Play searcher

Sistem pencarian aplikasi pada Google Play yang sesuai dengan kebutuhan pengguna.

Price:

	Everyone
X	Teen
X	Tools
	Entertainment

Gambar 4.24 Halaman pencarian

Halaman pada Gambar 4.24 merupakan tampilan awal dari sistem Google Play searcher, nama aplikasi untuk sistem pencarian aplikasi pada Google Play yang sesuai dengan kebutuhan pengguna. Pengguna memasukan kata pencarian

dan memasukan *filter* yang diperlukan/diinginkan untuk mendapatkan hasil yang sesuai, selanjutnya pengguna mengklik tombol cari untuk mencari aplikasi yang diinginkan.

4.2.4.2. Halaman hasil pencarian.

Google Play searcher

Masukan Pencarian

Cari

Price:

0

5

Your selected:

Tootls

X

Teen

X

	Everyone
X	Teen
X	Tools
	Entertainment

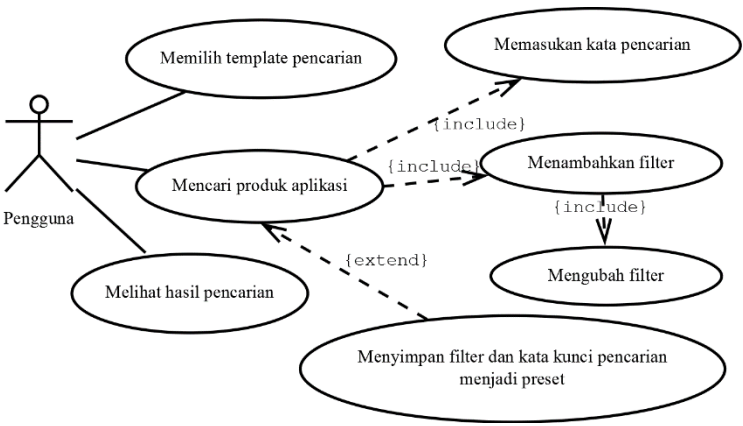
Icon	Google Fit
Icon	Nike Running
Icon	xx
Icon	xxxxx
Icon	xxxxxxx

Gambar 4.25 Halaman hasil pencarian

Halaman pada Gambar 4.25 merupakan tampilan untuk menampilkan hasil produk yang dicari dengan menambahkan *filter*. Informasi yang ditampilkan adalah *icon* produk dan nama produk aplikasi.

4.2.5. Use case

Use case dibuat berdasarkan antar muka sebagaimana yang dibuat pada bab 4.3.1. Berikut ini adalah gambar use case sistem secara keseluruhan.



Gambar 4.26 Use case diagram oleh pengguna

4.2.5.1. Daftar use case

Tabel 4.8 Daftar use case

Kode	Aktor	Nama
UCP-1	Pengguna	Memilih <i>template</i> pencarian
UCP-2	Pengguna	Mencari produk aplikasi
UCP-3	Pengguna	Melihat hasil pencarian
UCP-4	Pengguna	Memasukan kata pencarian
UCP-5	Pengguna	Menambahkan <i>filter</i>
UCP-6	Pengguna	Mengubah <i>filter</i>
UCP-7	Pengguna	Menyimpan <i>filter</i> dan kata kunci pencarian menjadi <i>template</i>

4.2.5.2. Deskripsi use case

Tabel 4.9 UCP-1: Memilih *template* pencarian

UCP-1: Memilih <i>template</i> pencarian
Normal Flow: pengguna memilih <i>template</i> pencarian, pengguna mengklik tombol submit/menekan <i>enter</i> pada <i>keyboard</i> milik pengguna, sistem menampilkan hasil pencarian dari <i>template</i> .
Alternate Flow: pengguna memilih <i>template</i> pencarian, pengguna tidak mengklik tombol submit/menekan <i>enter</i> pada <i>keyboard</i> milik pengguna, sistem tidak memproses pencarian dari <i>template</i> .

Tabel 4.10 UCP-2: Mencari produk aplikasi

UCP-2: Mencari produk aplikasi
Normal Flow: pengguna memasukan kata pencarian dikolom <i>input</i> , pengguna mengklik tombol submit/menekan <i>enter</i> pada <i>keyboard</i> milik pengguna, sistem menampilkan hasil pencarian.
Alternate Flow: pengguna memasukan kata pencarian dikolom <i>input</i> , pengguna tidak mengklik tombol submit/menekan <i>enter</i> pada <i>keyboard</i> milik pengguna, sistem tidak memproses pencarian dari kolom <i>input</i> .

Tabel 4.11 UCP-3: Melihat hasil aplikasi

UCP-3: Melihat hasil pencarian
Normal Flow: sistem menampilkan hasil pencarian, pengguna dapat melihat hasil dari pencarian yang dilakukan.

Alternate Flow: -

Tabel 4.12 UCP-4: Memasukan kata pencarian

UCP-4: Memasukan kata pencarian
Normal Flow: pengguna memasukan kata pencarian di kolom <i>input</i> .
Alternate Flow: -

Tabel 4.13 UCP-5: Menambah *filter*

UCP-5: Menambah <i>filter</i>
Normal Flow: pengguna menambahkan <i>filter</i> untuk menyaring hasil pencarian
Alternate Flow: -

Tabel 4.14 UCP-6: Mengubah *filter*

UCP-6: Mengubah <i>filter</i>
Normal Flow: pengguna mengubah <i>filter</i> yang telah dipilih, dengan cara menambahkan <i>filter</i> .
Alternate Flow: pengguna mengubah <i>filter</i> yang telah dipilih, dengan cara mengurangi <i>filter</i> .

Tabel 4.15 UCP-7: Menyimpan *filter* dan katakunci pencarian menjadi *template*

UCP-7: Menyimpan <i>filter</i> dan katakunci pencarian menjadi <i>template</i>
Normal Flow: sistem menampilkan hasil pencarian yang telah ditambahkan <i>filter</i> , pengguna mengisi kolom input nama <i>template</i> , pengguna mengklik tombol submit/menekan <i>enter</i> pada <i>keyboard</i> milik pengguna,

sistem menyimpan pencarian dan penyaringan untuk produk aplikasi ke basis data.

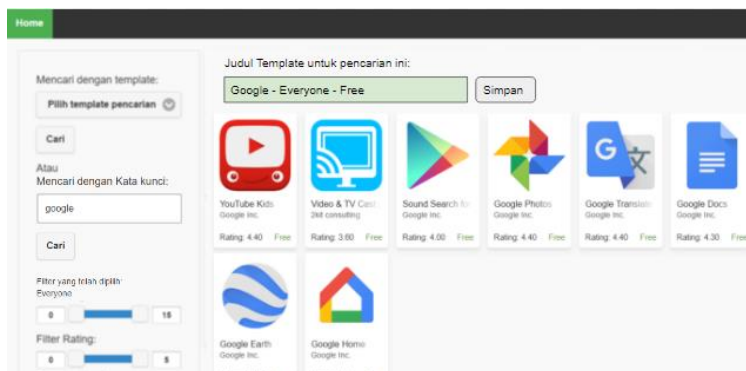
Alternate Flow: sistem menampilkan hasil pencarian yang telah ditambahkan *filter*, pengguna mengisi kolom input nama *template*, pengguna tidak mengklik tombol submit/menekan *enter* pada *keyboard* milik pengguna, sistem tidak memproses penyimpanan ke basis data.

4.3. Rancangan perbaikan usabilitas fitur pengelolaan

Perbaikan usabilitas fitur dapat dilakukan untuk mengelola pencarian dan penyaringan dengan menambahkan fitur pada aplikasi.

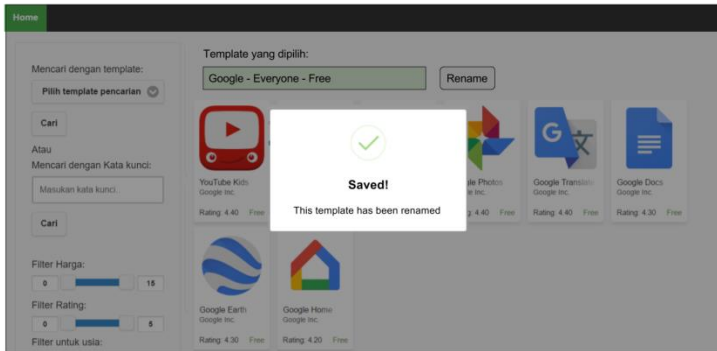
4.3.1. Pemberian nama *template* secara otomatis

Dengan menambahkan pemberian nama *template* secara otomatis pengguna tidak perlu memikirkan nama *template* jika ingin melakukan pemberian. Pengguna dapat tetap menggunakan *template*.



Gambar 4.27 Pemberian nama *template* secara otomatis

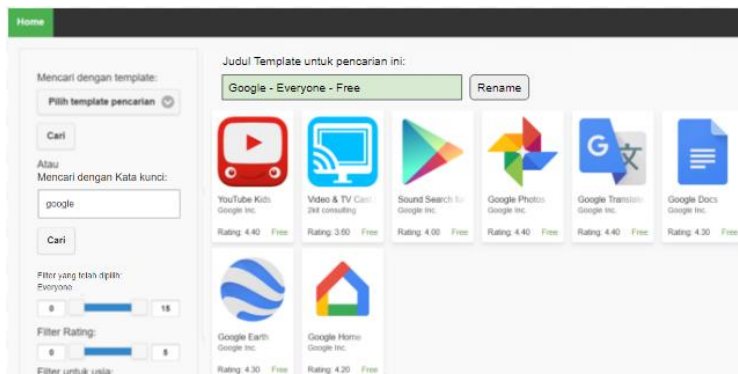
Pada Gambar 4.27 perbaikan pengelolaan usability fitur dilakukan dengan memasukan pencarian dan penyaringan yang dilakukan oleh pengguna mejadi judul *template*.



Gambar 4.28 Notifikasi pemberian judul *template*

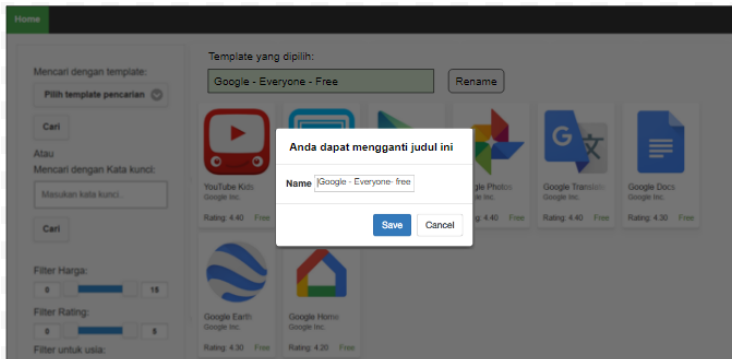
Ketika pengguna menekan tombol simpan, maka akan tampil notifikasi pemberian judul *template* telah tersimpan.

4.3.2. Pengubahan nama *template* pencarian



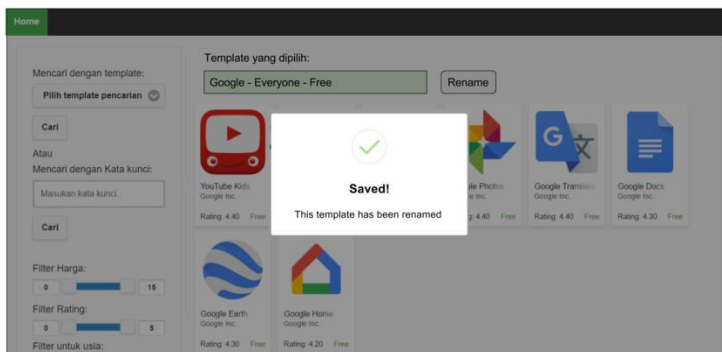
Gambar 4.29 Pengubahan nama template pencarian

pengelolaan usability fitur dilakukan dengan pengguna dapat melakukan perubahan judul pencarian *template*.



Gambar 4.30 Mengubah nama/judul pencarian *template*

Dengan menekan tombol *rename*, pengguna dapat mengubah judul *template* yang ada. Jika menekan tombol *save* maka judul yang baru akan tersimpan. Jika menekan tombol *cancel*.

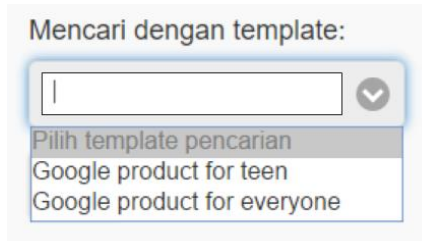


Gambar 4.31 Notifikasi pemberian judul *template*

Ketika pengguna menekan tombol simpan, maka akan tampil notifikasi pemberian judul *template* telah tersimpan.

4.3.3. Perancangan pencarian judul *template*

Dengan menambahkan pencarian judul *template* pengguna juga dapat dimudahkan dalam menemukan pencarian yang telah disimpan.



Gambar 4.32 Perancangan pencarian judul *template*

Pada Gambar 4.32 dirancang *searchable dropdown* untuk mencari *template* yang telah tersimpan dalam basis data. Dengan *searchable dropdown* data judul/nama *template* yang telah tersimpan dapat dicari oleh pengguna sehingga pengguna tidak perlu menelusuri satu per satu judul/nama *template* yang ada.

BAB V

IMPLEMENTASI

Pada bagian ini akan menjelaskan hasil implementasi yang telah dilakukan dalam tugas akhir terkait pengembangan aplikasi pada tiap aspek diteliti, yakni pencarian kebutuhan, pengambilan data, rancang bangun prototipe, dan validasi prototipe.

5.1. Lingkungan implementasi

Pada pengembangan aplikasi, peneliti menggunakan perangkat keras dengan spesifikasi seperti pada Tabel 5.1. Sedangkan untuk perangkat lunak yang digunakan dalam pengembangan aplikasi adalah seperti pada Tabel 5.2.

Tabel 5.1 Lingkungan implementasi (perangkat keras)

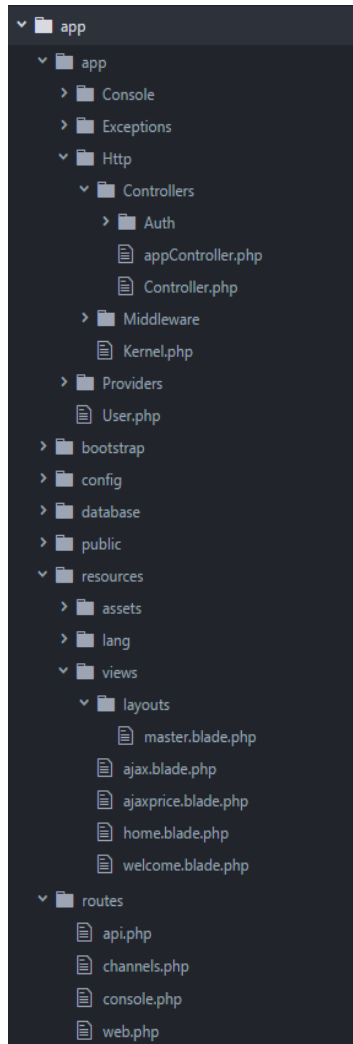
Perangkat keras	Spesifikasi
Laptop	Prosesor: AMD® APU A10-8700P 4 CPU Core
	RAM: 6GB
	Sistem Operasi Windows 10

Tabel 5.2 Lingkungan implementasi (perangkat lunak)

Perangkat lunak	Spesifikasi
Web server	Apache
Database server	MySQL
Bahasa	PHP
Framework	Laravel
Editor	Atom

5.2. Struktur direktori

Berikut adalah struktur direktori untuk pengembangan aplikasi pencarian dan penyaringan produk aplikasi pada Google Play.



Gambar 5.1 Struktur direktori aplikasi

5.3. Pengambilan data

Sebelum menampilkan data aplikasi, dilakukan pengambilan data dengan melakukan *web scraping*. *Web scraping*

dilakukan dengan menggunakan cURL (client Uniform Resource Locator) yang mengambil data dengan format json. Json yang berhasil diambil lalu di *decode*. Untuk mendapatkan nilai yang ada di dalam json, diperlukan wadah untuk menaruh nilai tersebut. Wadah tersebut dinamakan variabel. Sebelum data ditaruh di variabel, perlu diketahui terlebih dahulu nilai yang dibawa oleh json. Variabel yang sudah menampung nilai ditaruh ke *query* yang berfungsi memasukkan data ke dalam basis data.

```
<?php

$db_user = 'root';
$db_pass = "";
$db_name = 'googleplay_api';

// connect to database
$dsn      = 'mysql:dbname=' . $db_name .
';charset=utf8;host=localhost';
try {
    $dbh = new PDO( $dsn, $db_user, $db_pass );
} catch ( PDOException $e ) {
    echo 'Koneksi Gagal : ' . $e->getMessage();
}
?>
```

Gambar 5.2 Konfigurasi koneksi ke basis data

Kode pada Gambar 5.2 merupakan konfigurasi untuk koneksi ke basis data *googleplay_api*.

5.3.1. Pembuatan *scraper*

Pada bagian ini akan menjelaskan proses *scraper* yang akan mengambil data *list* dan data *app*. *List* merupakan istilah yang dibuat oleh peneliti, yaitu proses mengambil daftar aplikasi dari suatu koleksi. *App* juga merupakan istilah yang dibuat oleh peneliti, yaitu proses mengambil data aplikasi secara utuh

dari aplikasi. List dan app akan mengambil data dari Google Play API.

```
$service_url =
"https://gooplayapi.herokuapp.com/list/new_paid/$category/110/en/en/100/100";
$curl = curl_init($service_url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
//execute the session
$curl_response = curl_exec($curl);
//finish off the session
curl_close($curl);
$curl_json = json_decode($curl_response, true);
```

Gambar 5.3 Memasukan data ke proses cURL

Kode pada Gambar 5.3 merupakan lanjutan dari proses pengambilan data kategori. Kategori yang ada pada Google Play API dimasukkan ke proses cURL (*client URL*) untuk mendapatkan data berbentuk json. Data json yang didapat kemudian di *decode*. Untuk mengeluarkan nilai dari json yang telah di *decode* perlu dilakukan *looping*.

```
foreach ($curl_json as $val) {
    if(isset($val['url'])){
        $url = htmlspecialchars($val['url'], ENT_QUOTES);
    }
    else{
        $url = "";
    }
    if(isset($val['title'])){
        $title = htmlspecialchars($val['title'], ENT_QUOTES);
    }
    else{
        $title = "";
    }
}
```

```

    }
    if(isset($val['appId'])){
        $appId = htmlspecialchars($val['appId'],
ENT_QUOTES);
    }
    else{
        $appId = "";
    }
    if(isset($val['summary'])){
        $summary = htmlspecialchars($val['summary'],
ENT_QUOTES);
    }
    else{
        $summary = "";
    }
    if(isset($val['developer'])){
        $developer = htmlspecialchars($val['developer'],
ENT_QUOTES);
    }
    else{
        $developer = "";
    }
    if(isset($val['icon'])){
        $icon = htmlspecialchars($val['icon'], ENT_QUOTES);
    }
    else{
        $icon = "";
    }
    if(isset($val['score'])){
        $score = htmlspecialchars($val['score'],
ENT_QUOTES);
    }
    else{
        $score = "";
    }
    if(isset($val['price'])){
        $price = htmlspecialchars($val['price'],

```

```

ENT_QUOTES);
    }
    else{
        $price = "";
    }
    if(isset($val['free'])){
        $free = htmlspecialchars($val['free'], ENT_QUOTES);
    }
    else{
        $free = "";
    }
}

```

Gambar 5.4 Pengolahan data dari cURL

Kode pada Gambar 5.4 merupakan proses pengolahan data yang diambil dari cURL (*client* URL). Agar data json yang didapat tidak hanya berwujud data mentah, dilakukan konversi karakter data json yang belum terdefinisi menjadi entitas HTML dengan `htmlspecialchars`.

```

$query = "select appId from applications where
appId='".$sappid.'"";
$result1 = $dbh->query($query)->fetch();
if($result1==false){
    $table = 'applications'; // ubah ke nama tabel
    $field = "url`,`title`,`appId`,`summary`,`developer`,`icon`,`score`,`price`,`free`"; // kolomnya, kalo > 1 pisahkan pakai koma
    $val = "?,?,?,?,?,?,?,?"; // ini sesuai jumlah kolomnya,
    pakai koma
    $array = array( $url, $title, $sappid, $summary,
    $developer, $icon, $score, $price, $free ); // sesuai jumlah
    kolom juga
    $sth = $dbh->prepare( "INSERT INTO $table ($field)
VALUES ($val)" );
    $input = $sth->execute( $array );
}

```

```
}

```

Gambar 5.5 Memasukan data dari proses list

Kode pada Gambar 5.5 merupakan proses memasukan data dengan pengecekan kondisi, yaitu aplikasi yang sudah ada pada basis data tidak akan masuk lagi ke dalam basis data. Cara untuk mengeceknya adalah dengan mencocokkan appId. appId ini nantinya akan dijadikan sebagai *unique key*.

Lalu proses selanjutnya, yaitu mengambil data spesifik aplikasi secara utuh. Proses yang membedakan antara list dengan app adalah pemanggil untuk *looping*-nya tidak melalui kategori seperti hal-nya pada list, melainkan pada appId yang berfungsi mengambil data sisa yang belum terambil dari proses list. Lalu pembeda lainnya berada di data yang bersifat array, diperlukan pemilahan data. Berikut adalah potongan kode pengambilan data spesifik aplikasi di Google Play yang memiliki ekstensi php.

```
$query = "select appid from applications";
$result1 = $dbh->query($query)->fetchAll();
foreach($result1 as $r1){
    $appid_play = $r1['appid'];
}
```

Gambar 5.6 Mengambil data yang belum terambil dari proses list

Kode pada Gambar 5.6 adalah proses pengambilan data yang belum terambil dari proses list dengan mengacu pada appId produk aplikasi.

```
if(isset($vali['recentChanges'])){
    $recentChanges = htmlspecialchars(implode(" ",
    $vali['recentChanges']), ENT_QUOTES);
}
else if(strlen($vali['recentChanges'])==0){
    $recentChanges = "";
} else {
```

```

        $video =
htmlspecialchars($vali['recentChanges'], ENT_QUOTES);
    }

```

Gambar 5.7 Mengkonversi data yang didapat dari proses app

Kode pada Gambar 5.7 merupakan proses mengkonversi data yang didapat menjadi entitas HTML dengan htmlspecialchars dan memilah data ke array.

```

$query2 = "UPDATE applications SET minInstalls = '
$minInstalls', maxInstalls = '$maxInstalls', reviews =
'$reviews', developerEmail = '$developerEmail',
developerWebsite = '$developerWebsite', updated =
'$updated', genre = '$genre', genreId = '$genreId',
description = '$description', descriptionHTML =
'$descriptionHTML', offersIAP = '$offersIAP', video =
'$video', adSupported = '$adSupported', androidVersion =
'$androidVersion', contentRating = '$contentRating',
androidVersionText = '$androidVersionText',
recentChanges = '$recentChanges', preregister =
'$preregister' WHERE appId = '$appid_play'";

if($dbh->exec($query2))
{
    echo 'sukses';
}
else {
    echo 'gagal';
}

```

Gambar 5.8 Memasukan data yang sudah dikonversi ke basis data

Kode pada Gambar 5.8 diatas merupakan proses sistem dalam memasukan data yang didapat dan terkonversi dari pengambilan app pada Gambar 5.6 dan Gambar 5.7 ke dalam basis data.

5.4. Rancang bangun prototipe

Tahapan ini merupakan implementasi dari tahapan analisis dan perancangan ke dalam bentuk aplikasi sesuai dengan penjelasan lingkungan implementasi.

5.4.1. Implementasi halaman pencarian

Halaman pencarian produk pada Google Play berperan sebagai halaman utama dari aplikasi Google Play searcher. Pengguna memasukkan kata kunci pada kolom pencarian atau memilih *template* pencarian yang sudah disediakan dalam bentuk *dropdown*. Pengguna juga dapat menambahkan penyaringan pencarian untuk mendapatkan produk yang diinginkan. Selanjutnya sistem akan mencocokkan kata kunci pencarian dengan hasil pada basis data produk. Jika produk ditemukan, maka pengguna akan ditampilkan halaman hasil pencarian sesuai dengan kata kunci dengan tambahan penyaringan maupun tanpa penyaringan produk yang dicari. Namun, jika pencarian tidak sesuai dengan nama produk yang terdapat dalam basis data, maka sistem tidak akan menampilkan hasil apapun yang menunjukkan tidak ada produk yang ditemukan.

```
home.blade.php
```

```
<form class = "sidepanel" action="" method="GET">
  <select name="presetdrop">
    <option value="" disabled selected>Select a filtered
Search below</option>
    @foreach($tes as $r2)
      <option value="{{ $r2->title }}">{{ $r2->title
}}</option>
    @endforeach
  </select>
  <h4>OR</h4>
  <input type="text" name="submit" placeholder="Search
your apps.." value="{{ Input::get('submit') }}">
  <input class="ui-button ui-widget ui-corner-all"
type="submit" data-inline="true" value="Submit">
```

Gambar 5.9 *Template* dan kolom pencarian untuk halaman pencarian

Kode pada Gambar 5.9 digunakan untuk menampilkan pilihan *template* pencarian berupa *dropdown* serta menampilkan kolom untuk memasukan kata kunci pencarian.

home.blade.php
<pre> <div data-role="rangeslider"> <h4>Price:</h4> <input type="range" name="price-min" id="price-min" min="0" max="15" <?php if(isset(\$_GET['price-min'])) {?> value="{{Input::get('price-min')}}" <?php }else{ echo 'value="0"'; } ?>> <input type="range" name="price-max" id="price- max" min="0" max="15" <?php if(isset(\$_GET['price-max'])) {?> value="{{Input::get('price-max')}}" <?php }else{ echo 'value="15"'; } ?>> </div> </pre>

Gambar 5.10 Penyaringan *range slider* untuk *price* produk

Kode pada Gambar 5.10 merupakan masukan yang berbentuk *range slider* untuk memasukan penyaringan produk berdasarkan rentang harga yang dipilih.

```
home.blade.php
<div data-role="rangeslider">
  <h4>Rating:</h4>
  <input type="range" name="score-min" id="score-min"
min="0" max="5"
  <?php
  if(isset($_GET['score-min']))
  {?>
    value="{{ Input::get('score-min') }}"
  <?php
  }else{
    echo 'value="0"';
  }
  ?>>
  <input type="range" name="score-max" id="score-
max" min="0" max="5"
  <?php
  if(isset($_GET['score-max']))
  {?>
    value="{{ Input::get('score-max') }}"
  <?php
  }else{
    echo 'value="5"';
  }
  ?>>
</div>
```

Gambar 5.11 Penyaringan *range slider* untuk *rating* produk

Kode pada Gambar 5.11 merupakan masukan yang berbentuk *range slider* untuk memasukan penyaringan produk berdasarkan rentang rating yang dipilih.

```

home.blade.php
<h4>Age Rating</h4>
<input type="checkbox" name="Everyone" id="age1"
style="display: none" value="Everyone"
<?php if(isset($_GET['Everyone'])) echo
"checked='checked'"; ?>> <label for="age1"
>Everyone</label>
<input type="checkbox" name="Teen" id="age2"
style="display: none" value="Teen"
<?php if(isset($_GET['Teen'])) echo "checked='checked'";
?>> <label for="age2" >Teen</label>
<input type="checkbox" name="Mature" id="age3"
style="display: none" value="Mature 17+"
<?php if(isset($_GET['Mature'])) echo
"checked='checked'"; ?>> <label for="age3">Mature
17+</label>
<input type="checkbox" name="Adults" id="age4"
style="display: none" value="Adults onl"
<?php if(isset($_GET['Adults'])) echo
"checked='checked'"; ?>> <label for="age4">Adults
Only</label> </div>

```

Gambar 5.12 Penyaringan pilihan produk berdasarkan usia

Kode pada Gambar 5.12 merupakan masukan berbentuk *checkbox* yang dapat dipilih/ditambahkan pengguna sebagai penyaringan berdasarkan usia. Penyaringan dapat dipilih lebih dari 1 untuk memperluas hasil pencarian produk.

```
home.blade.php
```

```
<h4>Category</h4>
```

```
<input type="checkbox" name="Entertainment" id="cat1"
style="display: none" value="Entertainment"> <label
for="cat1">Entertainment</label>
```

```
<input type="checkbox" name="Audio" id="cat2"
style="display: none" value="Music"> <label
for="cat2">Music & Audio</label>
```

```
<input type="checkbox" name="Communication"
id="cat3" style="display: none" value="Communication">
<label for="cat3">Communication</label>
```

```
<input type="checkbox" name="Social" id="cat4"
style="display: none" value="Social"> <label
for="cat4">Social</label>
```

```
<input type="checkbox" name="Action" id="cat5"
style="display: none" value="Action"> <label
for="cat5">Action</label>
```

```
<input type="checkbox" name="Tools" id="cat6"
style="display: none" value="Tools"> <label
for="cat6">Tools</label>
```

```
<input type="checkbox" name="Word" id="cat7"
style="display: none" value="Word"> <label
for="cat7">Word</label>
```

```
<input type="checkbox" name="Shopping" id="cat8"
style="display: none" value="Shopping"> <label
for="cat8">Shopping</label>
```

```
<input type="checkbox" name="Simulation" id="cat9"
style="display: none" value="Simulation"> <label
for="cat9">Simulation</label>
```

```
<input type="checkbox" name="Puzzle" id="cat10"
style="display: none" value="Puzzle"> <label
for="cat10">Puzzle</label>
```

```
<input type="checkbox" name="Arcade" id="cat11"
style="display: none" value="Arcade"> <label
for="cat11">Arcade</label>
```

```
<input type="checkbox" name="Maps" id="cat12"
style="display: none" value="Maps"> <label
for="cat12">Maps & Navigation</label>
```

```

<input type="checkbox" name="Adventure" id="cat13"
style="display: none" value="Adventure"> <label
for="cat13">Adventure</label>
<input type="checkbox" name="Sports" id="cat14"
style="display: none" value="Sports"> <label
for="cat14">Sports</label>
<input type="checkbox" name="Racing" id="cat15"
style="display: none" value="Racing"> <label
for="cat15">Racing</label>
<input type="checkbox" name="Productivity" id="cat16"
style="display: none" value="Productivity"> <label
for="cat16">Productivity</label>
<input type="checkbox" name="Casual" id="cat17"
style="display: none" value="Casual"> <label
for="cat17">Casual</label>
<input type="checkbox" name="Board" id="cat18"
style="display: none" value="Board"> <label
for="cat18">Board</label>
<input type="checkbox" name="Personalization"
id="cat19" style="display: none" value="Personalization">
<label for="cat19">Personalization</label>
<input type="checkbox" name="Strategy" id="cat20"
style="display: none" value="Strategy"> <label
for="cat20">Strategy</label>
<input type="checkbox" name="Photography" id="cat21"
style="display: none" value="Photography"> <label
for="cat21">Photography</label>
<input type="checkbox" name="Music" id="cat22"
style="display: none" value="Music"> <label
for="cat22">Music</label>
<input type="checkbox" name="News" id="cat23"
style="display: none" value="News"> <label
for="cat23">News & Magazines</label>
<input type="checkbox" name="Finance" id="cat24"
style="display: none" value="Finance"> <label
for="cat24">Finance</label>
<input type="checkbox" name="Card" id="cat25"

```

```

style="display:      none"      value="Card">      <label
for="cat25">Card</label>
  <input  type="checkbox"    name="Travel"   id="cat26"
style="display:      none"      value="Travel">      <label
for="cat26">Travel & Local</label>
  <input  type="checkbox"    name="Food"     id="cat27"
style="display:      none"      value="Food">      <label
for="cat27">Food & Drink</label>
  <input  type="checkbox"    name="Lifestyle" id="cat28"
style="display:      none"      value="Lifestyle">  <label
for="cat28">Lifestyle</label>
  <input  type="checkbox"    name="Weather"  id="cat29"
style="display:      none"      value="Weather">  <label
for="cat29">Weather</label>
  <input  type="checkbox"    name="Video"    id="cat30"
style="display:      none"      value="Video">      <label
for="cat30">Video Players & Editors</label>
  <input  type="checkbox"    name="Books"    id="cat31"
style="display:      none"      value="Books">      <label
for="cat31">Books & Reference</label>
  <input  type="checkbox"    name="House"    id="cat32"
style="display:      none"      value="House">      <label
for="cat32">House & Home</label>
  <input  type="checkbox"    name="Vehicles" id="cat33"
style="display:      none"      value="Vehicles">  <label
for="cat33">Auto & Vehicles</label>
  <input  type="checkbox"    name="Health"   id="cat34"
style="display:      none"      value="Health">      <label
for="cat34">Health & Fitness</label>
  <input  type="checkbox"    name="Casino"   id="cat35"
style="display:      none"      value="Casino">      <label
for="cat35">Casino</label>
  <input  type="checkbox"    name="Education" id="cat36"
style="display:      none"      value="Education">  <label
for="cat36">Education</label>
  <input  type="checkbox"    name="Role"     id="cat37"
style="display:      none"      value="Role">      <label

```

```
for="cat37">Role Playing</label>
```

Gambar 5.13 Penyaringan pilihan produk berdasarkan kategori produk

Kode pada Gambar 5.13 merupakan masukan berbentuk *checkbox* yang dapat dipilih/ditambahkan pengguna sebagai penyaringan berdasarkan kategori. Penyaringan dapat dipilih lebih dari 1 untuk memperluas hasil pencarian produk.

```
home.blade.php
```

```

<input id="ver1" name="VARY" type="checkbox"
value="VARY"> <label for="ver1">VARY</label>
  <input id="ver2" name="v50" type="checkbox"
value="5.0"> <label for="ver2">5.0</label>
  <input id="ver3" name="v44" type="checkbox"
value="4.4"> <label for="ver3">4.4</label>
  <input id="ver4" name="v43" type="checkbox"
value="4.3"> <label for="ver4">4.3</label>
  <input id="ver5" name="v42" type="checkbox"
value="4.2"> <label for="ver5">4.2</label>
  <input id="ver6" name="v41" type="checkbox"
value="4.1"> <label for="ver6">4.1</label>
  <input id="ver7" name="v403" type="checkbox"
value="4.0.3"> <label for="ver7">4.0.3</label>
  <input id="ver8" name="v40" type="checkbox"
value="4.0"> <label for="ver8">4.0</label>
  <input id="ver9" name="v32" type="checkbox"
value="3.2"> <label for="ver9">3.2</label>
  <input id="ver10" name="v31" type="checkbox"
value="3.1"> <label for="ver10">3.1</label>
  <input id="ver11" name="v30" type="checkbox"
value="3.0"> <label for="ver11">3.0</label>
  <input id="ver12" name="v233" type="checkbox"
value="2.3.3"> <label for="ver12">2.3.3</label>
  <input id="ver13" name="v23" type="checkbox"
value="2.3"> <label for="ver13">2.3</label>
  <input id="ver14" name="v22" type="checkbox"
value="2.2"> <label for="ver14">2.2</label>
  <input id="ver15" name="v21" type="checkbox"
value="2.1"> <label for="ver15">2.1</label>
  <input id="ver16" name="v201" type="checkbox"
value="2.0.1"> <label for="ver16">2.0.1</label>
  <input id="ver17" name="v20" type="checkbox"
value="2.0"> <label for="ver17">2.0</label>
  <input id="ver18" name="v16" type="checkbox"
value="1.6"> <label for="ver18">1.6</label>
  <input id="ver19" name="v15" type="checkbox"

```

```

value="1.5"> <label for="ver19">1.5</label>
<input id="ver20" name="v11" type="checkbox"
value="1.1"> <label for="ver20">1.1</label>
<input id="ver21" name="v10" type="checkbox"
value="1.0"> <label for="ver21">1.0</label>

```

Gambar 5.14 Penyaringan pilihan produk berdasarkan versi android

Kode pada Gambar 5.14 merupakan masukan berbentuk *checkbox* yang dapat dipilih/ditambahkan pengguna sebagai penyaringan berdasarkan versi android. Penyaringan dapat dipilih lebih dari 1 untuk memperluas hasil pencarian produk.

```

home.blade.php
<input id="ear1" name="ear1" type="checkbox" value="1">
<label for="ear1">early access</label>
<input id="ear2" name="ear2" type="checkbox" value="0">
<label for="ear2">tidak early access</label>

```

Gambar 5.15 Penyaringan pilihan produk yang dapat diakses secara *early*

Kode pada Gambar 5.15 merupakan masukan berbentuk *checkbox* yang dapat dipilih/ditambahkan pengguna sebagai penyaringan produk aplikasi yang dapat diakses *early* (belum rilis secara global).

```

home.blade.php
<input id="ads1" name="ads1" type="checkbox" value="1">
<label for="ads1">ada iklan</label>
<input id="ads2" name="ads2" type="checkbox" value="0">
<label for="ads2">tidak ada iklan</label>

```

Gambar 5.16 Penyaringan pilihan produk yang memiliki iklan dalam aplikasi

Kode pada merupakan masukan berbentuk *checkbox* yang dapat dipilih/ditambahkan pengguna sebagai penyaringan produk aplikasi yang terdapat iklan didalamnya.

```

appController.php
if(!empty($_GET['presetdrop'])){
    $presetdrop = $_GET['presetdrop'];
    $resdrop = DB::select("SELECT * FROM
filter_preset WHERE title LIKE '%".$presetdrop."%'");
    $database = DB::select("SELECT * FROM
filter_preset WHERE title LIKE '%zzzzzzzzzzzzzz%'");
    $tes = DB::table("filter_preset")->get();
    return view('home', ['database' => $database, 'tes' =>
$tes, 'resdrop' => $resdrop]);
}
if (isset($_GET['presettile'])) {
    $database = DB::select("SELECT ap.icon, ap.title,
ap.price, ap.score, ap.developer FROM applications ap
LEFT JOIN applications_facts af ON
af.app_id=ap.id
WHERE (ap.title LIKE '%$presetnamex%'
OR ap.developer LIKE '%$presetnamex%')
AND ($agex OR $genrex OR $andversionx OR
$adsupx OR $preregx
OR (af.facet_id = 8 AND (af.value <=
$presetscoredx)) OR (af.facet_id = 9 AND (af.value <=
$presetrevmaxx))
OR (af.facet_id = 7 AND (af.value <=
$presetinsmaxx)) OR (af.facet_id = 4 AND (af.value
BETWEEN $presetpminx AND $presetpmaxx)))
GROUP BY ap.id, ap.icon, ap.title,
ap.price, ap.score, ap.developer");
    $tes = DB::table("filter_preset")->get();
    return view('home', ['database' => $database, 'tes' =>
$tes]);
}

```

Gambar 5.17 *Query pemprosesan pencarian dengan template*

Kode pada Gambar 5.17 merupakan *query* pemprosesan masukan dari *dropdown template* yang akan menghasilkan data untuk ditampilkan pada halaman hasil pencarian.

appController.php
<pre> \$database = DB::select("SELECT ap.icon, ap.title, ap.price, ap.score, ap.developer FROM applications ap LEFT JOIN applications_facts af ON af.app_id=ap.id WHERE (ap.title LIKE '%\$term%' OR ap.developer LIKE '%\$term%') AND (\$age OR \$genre OR \$andversion OR \$adsup OR \$prereg OR (af.facet_id = 8 AND (af.value <= \$scored)) OR (af.facet_id = 9 AND (af.value <= \$revmax)) OR (af.facet_id = 7 AND (af.value <= \$insmax)) OR (af.facet_id = 4 AND (af.value BETWEEN \$pricemin AND \$pricemax))) GROUP BY ap.id, ap.icon, ap.title, ap.price, ap.score, ap.developer"); \$tes = DB::table("filter_preset")->get(); return view('home', ['database' => \$database, 'tes' => \$tes]); </pre>

Gambar 5.18 *Query pemprosesan pencarian dengan kolom*

Kode pada Gambar 5.18 merupakan *query* pemprosesan masukan dari kolom kata kunci yang akan menghasilkan data untuk ditampilkan pada halaman hasil pencarian.

appController.php
<pre> if (!isset(\$_GET['submit']) AND !isset(\$_GET['presetttitle'])){ \$database = DB::table("applications")->paginate(60); </pre>

```

$tes = DB::table("filter_preset")->get();
$resdrop = DB::table("filter_preset")->get();
return view('home', ['database' => $database, 'tes' =>
$tes, 'resdrop' => $resdrop,]);
}

```

Gambar 5.19 Query untuk memproses data saat tidak dilakukan pencarian

Kode pada Gambar 5.19 merupakan *query* untuk mengambil data dari basis data jika sistem tidak melakukan pencarian.

5.4.2. Implementasi halaman hasil pencarian

Halaman hasil berisikan hasil pencarian produk yang terdiri dari *icon* produk, nama produk, pengembang produk, *rating* dari produk, serta harga produk. Pada halaman ini juga tersedia opsi untuk menyimpan pencarian dan penyaringan yang telah dilakukan. Penyimpanan ini dimasukkan ke basis data sebagai *template* pencarian untuk memudahkan pengguna lain saat mencari produk dengan kata kunci dan penyaringan yang sama. Pengguna harus memberikan judul *template* untuk dapat menyimpan pencarian dan penyaringan yang telah dilakukan. Apabila tidak diberikan judul, sistem tidak akan memasukan kata kunci dan penyaringan yang telah dilakukan pengguna ke basis data.

```

home.blade.php
<div id="tempel">
<?php
foreach ($database as $r1) {
    echo "<div class='panel'>";
    echo "<div class='icon'>";
    echo "<img width='130' src='\".$r1->icon.\"> ";
    echo "</div>";
    echo "<div class='title'>";
    echo "\". $r1->title;
    echo "<span class='paragraph-end'></span>";

```

```

echo "</div>";
echo "<div class='devname'>";
echo "'". $r1->developer;
echo "<span class='paragraph-end'></span>";
echo "</div>";
echo "<br>";
echo "<div class='score'>";
echo " Rating: '". $r1->score;
echo "</div>";
echo "<div class='price'>";
if($r1->price != '0'){
    echo "'". $r1->price;
}else{
    echo "Free";
}
echo "</div>";
echo "</div>";
}
?>
</div>

<?php
if      (!isset($_GET['submit']))          AND
!isset($_GET['presettitle'])){
?>
    <br>
    <center>
    <div class="pagination">
        { { $database->links() } }
    </div>
    </center>
<?php
}

```

Gambar 5.20 Menampilkan hasil pencarian

Kode pada Gambar 5.20 merupakan pemrosesan untuk menampilkan hasil pencarian dengan *template*, dengan kolom kata kunci, maupun saat tidak dilakukan pencarian. Tampilan dari hasil kode ini dimuat pada halaman hasil yang dibenahi dengan CSS sehingga dapat dilihat dengan mudah oleh pengguna. Link *paggination* akan muncul jika jumlah data yang tampil melebihi 60 data.

Pada halaman hasil pencarian juga diterapkan javascript + ajax untuk menampilkan pencarian dengan penyaringan menggunakan *range slider*.

home.blade.php

```
$(window).load(function(){
    $("#score-max").on("slidestop", function(e){
        var min = document.getElementById('score-min').value;
        var max = document.getElementById('score-
max').value;

        $.ajax({
            url: 'ajax',
            type: 'GET',
            data: {"score-min":min,"score-max":max},
            success: function(response){
                // alert(response);
                $("#tempel").html(response);
            }
        });
    });
    $("#score-min").on("slidestop", function(e){
        var min = document.getElementById('score-min').value;
        var max = document.getElementById('score-
max').value;

        $.ajax({
            url: 'ajax',
```

```

    type: 'GET',
    data: { "score-min":min,"score-max":max },
    success: function(response){
        // alert(response);
        $("#tempel").html(response);
    }
    });
});
$("#price-min").on("slidestop", function(e){
    var min = document.getElementById('price-min').value;
    var    max    =    document.getElementById('price-
max').value;

    $.ajax({
        url: 'ajaxprice',
        type: 'GET',
        data: { "price-min":min,"price-max":max },
        success: function(response){
            // alert(response);
            $("#tempel").html(response);
        }
    });
});
$("#price-max").on("slidestop", function(e){
    var min = document.getElementById('price-min').value;
    var    max    =    document.getElementById('price-
max').value;

    $.ajax({
        url: 'ajaxprice',
        type: 'GET',
        data: { "price-max":min,"price-max":max },
        success: function(response){
            // alert(response);
            $("#tempel").html(response);
        }
    });
});

```

```

    });
  });
</script>

```

Gambar 5.21 Javascript dan ajax untuk menampilkan hasil penyaringan dengan range slider.

```

appController.php
public function ajax() {
    $scoremin = $_GET['score-min'];
    $scoremax = $_GET['score-max'];

    $database = DB::select("SELECT ap.icon, ap.title,
ap.price, ap.score, ap.developer FROM applications ap
LEFT JOIN applications_facts af ON af.app_id=ap.id
WHERE (af.facet_id = 8 AND (af.value BETWEEN
$scoremin AND $scoremax)) ");

    foreach ($database as $r1) {
        $hasil = "<div class='panel'>";
        $hasil .= "<div class='icon'>";
        $hasil .= "<img width='130' src='". $r1->icon. "'> ";
        $hasil .= "</div>";
        $hasil .= "<div class='title'>";
        $hasil .= "'". $r1->title;
        $hasil .= "<span class='paragraph-end'></span>";
        $hasil .= "</div>";
        $hasil .= "<div class='devname'>";
        $hasil .= "'". $r1->developer;
        $hasil .= "<span class='paragraph-end'></span>";
        $hasil .= "</div>";
        $hasil .= "<br>";
        $hasil .= "<div class='score'>";
        $hasil .= " Rating: ". $r1->score;
        $hasil .= "</div>";
        $hasil .= "<div class='price'>";

```

```

        if($r1->price != '0'){
            $hasil .= "".$r1->price;
        }else{
            $hasil .= "Free";
        }
        $hasil .= "</div>";
        $hasil .= "</div>";
        echo $hasil;
    }
}

```

Gambar 5.22 *Query untuk memproses penyaringan range slider rating*

appController.php

```

public function ajaxprice(){
    $pmin = $_GET["price-min"];
    if($pmin == '0'){
        $pricemin = $_GET["price-min"];
        $pricemax = $_GET["price-max"];
    }else{
        $pricemin = '$'.$_GET["price-min"];
        $pricemax = '$'.$_GET["price-max"];
    }

    $database = DB::select("SELECT ap.icon, ap.title,
ap.price, ap.score, ap.developer FROM applications ap
LEFT JOIN applications_facts af ON af.app_id=ap.id
WHERE (af.facet_id = 4 AND (af.value BETWEEN
'".$pricemin."' AND '".$pricemax."')) ");

    foreach ($database as $r1) {
        $hasil = "<div class='panel'>";
        $hasil .= "<div class='icon'>";
        $hasil .= "<img width='130' src='".$r1->icon."> ";
        $hasil .= "</div>";
        $hasil .= "<div class='title'>";
        $hasil .= "".$r1->title;
    }
}

```



```

$hasil .= "<span class='paragraph-end'></span>";
$hasil .= "</div>";
$hasil .= "<div class='devname'>";
$hasil .= "".$r1->developer;
$hasil .= "<span class='paragraph-end'></span>";
$hasil .= "</div>";
$hasil .= "<br>";
$hasil .= "<div class='score'>";
$hasil .= " Rating: ".$r1->score;
$hasil .= "</div>";
$hasil .= "<div class='price'>";
if($r1->price != '0'){
    $hasil .= "".$r1->price;
}else{
    $hasil .= "Free";
}
$hasil .= "</div>";
$hasil .= "</div>";
echo $hasil;
}
}

```

Gambar 5.23 Query untuk memproses penyaringan *range slider rating*

Kode pada Gambar 5.21 merupakan javascript + ajax untuk memproses penyaringan menggunakan *range slider* (penyaringan *rating* atau *price*). Untuk penyaringan *rating*, kode akan diproses ke *function* di Controller yang ditunjukkan pada Gambar 5.22. Sedangkan untuk penyaringan *price*, kode akan diproses ke *function* di Controller yang ditunjukkan pada Gambar 5.23. Untuk penyaringan *rating*, setelah kode diproses di *function*, sistem akan menampilkan hasilnya pada *view* `ajax.blade.php` yang berisi kode seperti pada Gambar 5.20. Untuk penyaringan *price*, kode akan menampilkan hasilnya pada *view* `ajaxprice.blade.php` yang berisi kode seperti pada Gambar 5.20.

Ajax digunakan untuk mengambil tampilan pada `ajax.blade.php` dan `ajaxprice.blade.php` yang dimuat langsung pada halaman hasil pencarian (`home.blade.php`).

Pada halaman hasil pencarian juga terdapat kolom masukan judul *template* dan tombol *save* yang akan muncul jika pengguna melakukan pencarian dengan menambahkan satu atau lebih penyaringan pada halaman pencarian.

home.blade.php		
<input	type="hidden"	name="saver1"
value="{{ Input::get('Everyone') }}">		
<input	type="hidden"	name="saver2"
value="{{ Input::get('Teen') }}">		
<input	type="hidden"	name="saver3"
value="{{ Input::get('Mature') }}">		
<input	type="hidden"	name="saver4"
value="{{ Input::get('Adults') }}">		
<input	type="hidden"	name="saver5"
value="{{ Input::get('Entertainment') }}">		
<input	type="hidden"	name="saver6"
value="{{ Input::get('Audio') }}">		
<input	type="hidden"	name="saver7"
value="{{ Input::get('Communication') }}">		
<input	type="hidden"	name="saver8"
value="{{ Input::get('Social') }}">		
<input	type="hidden"	name="saver9"
value="{{ Input::get('Action') }}">		
<input	type="hidden"	name="saver10"
value="{{ Input::get('Tools') }}">		
<input	type="hidden"	name="saver11"
value="{{ Input::get('Word') }}">		
<input	type="hidden"	name="saver12"
value="{{ Input::get('Shopping') }}">		
<input	type="hidden"	name="saver13"
value="{{ Input::get('Simulation') }}">		

<input type="hidden"	name="saver14"
value="{{ Input::get('Arcade') }}">	
<input type="hidden"	name="saver15"
value="{{ Input::get('Maps') }}">	
<input type="hidden"	name="saver16"
value="{{ Input::get('Adventure') }}">	
<input type="hidden"	name="saver17"
value="{{ Input::get('Sports') }}">	
<input type="hidden"	name="saver18"
value="{{ Input::get('Racing') }}">	
<input type="hidden"	name="saver19"
value="{{ Input::get('Productivity') }}">	
<input type="hidden"	name="saver20"
value="{{ Input::get('Casual') }}">	
<input type="hidden"	name="saver21"
value="{{ Input::get('Board') }}">	
<input type="hidden"	name="saver22"
value="{{ Input::get('Personalization') }}">	
<input type="hidden"	name="saver23"
value="{{ Input::get('Strategy') }}">	
<input type="hidden"	name="saver24"
value="{{ Input::get('Photography') }}">	
<input type="hidden"	name="saver25"
value="{{ Input::get('Music') }}">	
<input type="hidden"	name="saver26"
value="{{ Input::get('News') }}">	
<input type="hidden"	name="saver27"
value="{{ Input::get('Finance') }}">	
<input type="hidden"	name="saver28"
value="{{ Input::get('Card') }}">	
<input type="hidden"	name="saver29"
value="{{ Input::get('Travel') }}">	
<input type="hidden"	name="saver30"
value="{{ Input::get('Food') }}">	
<input type="hidden"	name="saver31"
value="{{ Input::get('Lifestyle') }}">	
<input type="hidden"	name="saver32"

value="{{ Input::get('Weather') }}">	
<input <="" td="" type="hidden"/> <td>name="saver33"</td>	name="saver33"
value="{{ Input::get('Video') }}">	
<input <="" td="" type="hidden"/> <td>name="saver34"</td>	name="saver34"
value="{{ Input::get('Books') }}">	
<input <="" td="" type="hidden"/> <td>name="saver35"</td>	name="saver35"
value="{{ Input::get('House') }}">	
<input <="" td="" type="hidden"/> <td>name="saver36"</td>	name="saver36"
value="{{ Input::get('Vehicles') }}">	
<input <="" td="" type="hidden"/> <td>name="saver37"</td>	name="saver37"
value="{{ Input::get('Health') }}">	
<input <="" td="" type="hidden"/> <td>name="saver38"</td>	name="saver38"
value="{{ Input::get('Casino') }}">	
<input <="" td="" type="hidden"/> <td>name="saver39"</td>	name="saver39"
value="{{ Input::get('Education') }}">	
<input <="" td="" type="hidden"/> <td>name="saver40"</td>	name="saver40"
value="{{ Input::get('Role') }}">	
<input <="" name="saver41" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('VARY') }}">	
<input <="" name="saver42" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v50') }}">	
<input <="" name="saver43" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v44') }}">	
<input <="" name="saver44" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v43') }}">	
<input <="" name="saver45" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v42') }}">	
<input <="" name="saver46" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v41') }}">	
<input <="" name="saver47" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v403') }}">	
<input <="" name="saver48" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v40') }}">	
<input <="" name="saver49" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v32') }}">	
<input <="" name="saver50" td=""/> <td>type="hidden"</td>	type="hidden"
value="{{ Input::get('v31') }}">	

```

    <input      name="saver51"      type="hidden"
value="{{ Input::get('v30') }}">
    <input      name="saver52"      type="hidden"
value="{{ Input::get('v233') }}">
    <input      name="saver53"      type="hidden"
value="{{ Input::get('v23') }}">
    <input      name="saver54"      type="hidden"
value="{{ Input::get('v22') }}">
    <input      name="saver55"      type="hidden"
value="{{ Input::get('v21') }}">
    <input      name="saver56"      type="hidden"
value="{{ Input::get('v201') }}">
    <input      name="saver57"      type="hidden"
value="{{ Input::get('v20') }}">
    <input      name="saver58"      type="hidden"
value="{{ Input::get('v16') }}">
    <input      name="saver59"      type="hidden"
value="{{ Input::get('v15') }}">
    <input      name="saver60"      type="hidden"
value="{{ Input::get('v11') }}">
    <input      name="saver61"      type="hidden"
value="{{ Input::get('v10') }}">

    <input      name="saver62"      type="hidden"
value="{{ Input::get('ads1') }}">
    <input      name="saver63"      type="hidden"
value="{{ Input::get('ads2') }}">

    <input      name="saver64"      type="hidden"
value="{{ Input::get('ear1') }}">
    <input      name="saver65"      type="hidden"
value="{{ Input::get('ear2') }}">
    <br>
    Filter title: <input type="text" name="preset"
value="{{ Input::get('preset') }}">
    <button type="submit" name="Savers" data-
inline="true" value="Savers"> Save </button>

```

Gambar 5.24 Masukan untuk menyimpan *template* pencarian

Kode pada Gambar 5.24 merupakan kolom untuk memasukan judul *template* dan tombol *save* yang akan muncul jika pengguna memilih pencarian dengan menambahkan penyaringan. Kode *hidden input* digunakan sebagai masukan yang mengambil *value* dari pencarian dan penyaringan yang akan diproses oleh *query* penyimpanan *template* kedalam basis data.

```
appController.php
$presetf = DB::table('filter_preset')->insertGetId(
    ['title' => $presetname, 'searchedname' => $term,
    'pmin' => $pricemin
    , 'pmax' => $pricemax, 'smin' => $scoremin, 'smax'
=> $scoremax
    , 'value1' => $eone, 'value2' => $teen, 'value3' =>
$mature17
    , 'value4' => $aonly, 'value5' => $sentment, 'value6'
=> $audio
    , 'value7' => $comm, 'value8' => $social, 'value9' =>
$act
    , 'value10' => $tools, 'value11' => $word, 'value12'
=> $shop
    , 'value13' => $sim, 'value14' => $arcade, 'value15'
=> $mapnav
    , 'value16' => $advent, 'value17' => $sport, 'value18'
=> $race
    , 'value19' => $productive, 'value20' => $casual,
'value21' => $boardg
    , 'value22' => $persona, 'value23' => $strat, 'value24'
=> $photo
    , 'value25' => $music, 'value26' => $news, 'value27'
=> $finance
    , 'value28' => $card, 'value29' => $travel, 'value30'
=> $food
    , 'value31' => $life, 'value32' => $weather, 'value33'
```

```

=> $vid
    , 'value34' => $book, 'value35' => $house, 'value36'
=> $vehicle
    , 'value37' => $health, 'value38' => $casino ,
'value39' => $edu
    , 'value40' => $role, 'value41' => $varyx, 'value42' =>
$v50x
    , 'value43' => $v44x, 'value44' => $v43x, 'value45'
=> $v42x
    , 'value46' => $v41x, 'value47' => $v403x, 'value48'
=> $v40x
    , 'value49' => $v32x, 'value50' => $v31x, 'value51'
=> $v30x
    , 'value52' => $v233x, 'value53' => $v23x, 'value54'
=> $v22x
    , 'value55' => $v21x, 'value56' => $v201x, 'value57'
=> $v20x
    , 'value58' => $v16x, 'value59' => $v15x, 'value60'
=> $v11x
    , 'value61' => $v10x, 'value62' => $ads1x, 'value63'
=> $ads2x
    , 'value64' => $ear1x, 'value65' => $ear2x]);

    $database      =      DB::table("applications")-
>paginate(60);
    $tes = DB::table("filter_preset")->get();
    return view('home', ['database' => $database,
'presetf' => $presetf, 'tes' => $tes]);

```

Gambar 5.25 Query untuk menyimpan masukan sebagai *template* pencarian kedalam basis data

Kode pada Gambar 5.25 merupakan *query* untuk menyimpan data yang telah dimasukan pada kolom *hidden input* kedalam basis data sebagai *template* pencarian produk.

5.4.3. Implementasi perbaikan usabilitas fitur pengelolaan

Dari rancangan perbaikan usabilitas fitur pengelolaan dilakukan pengembangan aplikasi dengan menambahkan fitur kedalam kode aplikasi.

```

if (isset($_GET["Everyone"])) {
    $sn[] = "- Everyone";
}
if (isset($_GET["Teen"])) {
    $sn[] = "- Teen";
}
if (isset($_GET["Mature"])) {
    $sn[] = "- Mature 17+";
}
if (isset($_GET["Adults"])) {
    $sn[] = "- Adults Only";
}

<input type="text" name="preset" placeholder="Judul
template pencarian.." value="
{{Input::get('submit')}} <?php echo htmlspecialchars($sne);
?>">

<button onclick="myFunction()" type="submit"
name="Savers" data-inline="true" value="Savers"> simpan
</button>

</div>

```

Gambar 5.26 kode pemberian nama *template* secara otomatis

Kode pada Gambar 5.26 merupakan *input* dengan mengambil nilai data dari penyaringan yang dipilih yang disimpan sementara kedalam *array*. Jika terdapat lebih dari satu penyaringan yang dipilih, *array* akan menyisipkan tanda *minus* (-) sebagai pemisah setiap penyaringan yang dipilih. Kemudian jika tersimpan akan muncul notifikasi bahwa data telah tersimpan.

Halaman ini sengaja dikosongkan

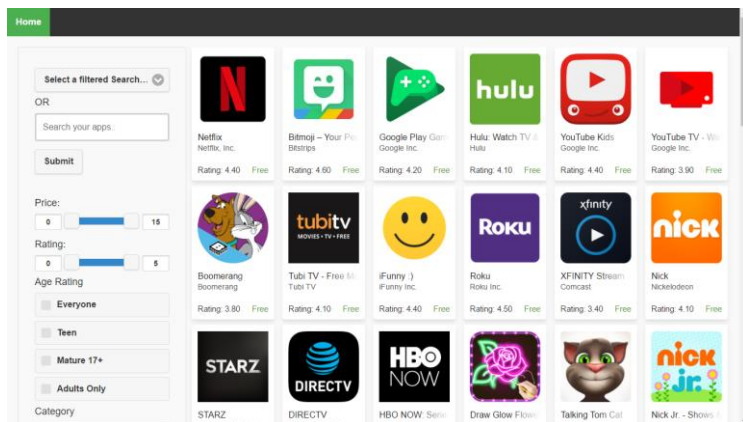
BAB VI

HASIL DAN PEMBAHASAN

6.1. Hasil prototipe aplikasi

Hasil prototipe aplikasi berupa antar muka pengguna yang siap dipakai dengan kapabilitas terbatas hanya pada fitur pencarian produk dan penyaringan serta melihat hasil pencarian produk yang dipilih oleh pengguna. Terdapat dua tampilan antar muka antara lain halaman pencarian dan halaman hasil pencarian yang dilebur menjadi satu halaman. Berikut adalah penjelasan mengenai masing-masing fitur halaman.

6.1.1. Halaman pencarian

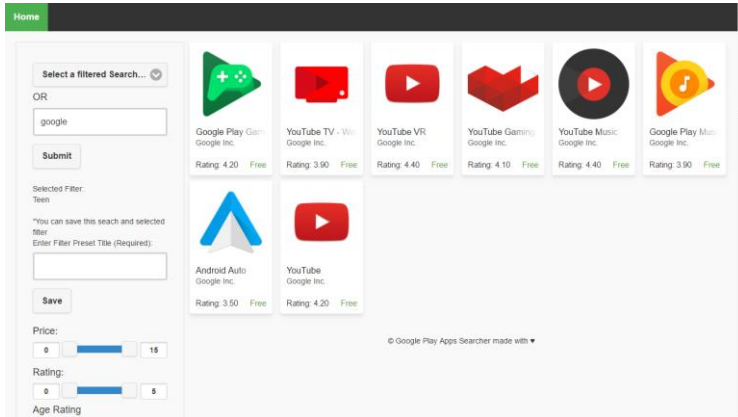


Gambar 6.1 Halaman pencarian produk

Halaman pada Gambar 6.1 merupakan halaman awal dari sistem Google Play searcher, nama aplikasi sistem pencarian dan penyaringan untuk memudahkan pengguna dalam mencari produk pada Google Play. Pengguna dapat memilih untuk menggunakan *template* pencarian atau memasukkan kata kunci untuk menemukan produk aplikasi yang diinginkan. Pengguna juga dapat menambahkan penyaringan pencarian berupa *range*

slider maupun *checkbox* untuk mempersempit pencarian dari produk yang diinginkan.

6.1.2. Halaman hasil pencarian



Gambar 6.2 Halaman hasil pencarian

Halaman pada Gambar 6.2 merupakan tampilan untuk menampilkan daftar produk sesuai dengan pilihan *template* atau kata kunci, baik dengan penyaringan maupun tanpa penyaringan. Data yang ditampilkan adalah daftar produk berupa *icon*, nama, pengembang, *rating*, dan harga dari produk. Pengguna tidak dapat memilih produk untuk melihat lebih jauh informasi produk. Karena lingkup sistem hanya sampai pada tahap menampilkan saja. Pengguna juga dapat menyimpan pencarian beserta penyaringan yang dilakukan kedalam *template* pada basis data dengan cara memasukkan judul *template* dan menekan tombol *save*.

6.1.3. Hasil pembuatan penyaringan

Hasil pembuatan penyaringan adalah *filter* yang dibentuk dari metadata yang ada pada basis data. Terdapat 16 meta data yang dapat dijadikan penyaringan.

Kriteria metadata dapat menjadi *filter* adalah datanya berbentuk *range* (rentang), *list* (daftar), waktu/tanggal, dan Expression [16]. Dalam pembuatan *filter* produk jika data yang digunakan dapat menciptakan susunan data baru dan memuat data yang digunakan tersebut, maka data tersebut juga bisa menjadi filter. Ketika data tidak bisa menciptakan susunan data yang baru data tersebut sebaiknya tidak dijadikan *filter* [17]. Berikut adalah metadata yang dapat dijadikan penyaringan:

Tabel 6.1 Metadata yang dapat dijadikan penyaringan

No	Nama kolom	Bisa menjadi <i>filter</i>	Alasan	Keterangan
1	id	Tidak bisa	Merupakan id tiap data	
2	title	Tidak bisa	Menjadi kolom pencarian dengan kata kunci	Nama dari aplikasi
3	summary	Tidak bisa	Hanya berupa deskripsi aplikasi	Outline/abstrak/tag dari aplikasi
4	icon	Tidak bisa	Gambar/logo/simbol dari aplikasi	
5	price	Bisa	Harga jual dari aplikasi	<i>Range</i>
6	free	Bisa	Memilah mana aplikasi yang gratis dan berbayar	Poin ini digabung menjadi satu dengan <i>filter price</i>
7	minInstalls	Bisa	Instalasi	<i>Range</i>

			minimal dari aplikasi	
8	maxInstall s	Bisa	Installasi maksimal dari aplikasi	Poin ini digabung menjadi satu dengan <i>filter MinInstall</i>
9	score	Bisa	Memilah rating bintang dari produk aplikasi	<i>Range</i>
10	reviews	Bisa	Jumlah review yang diberikan untuk aplikasi	<i>Range</i>
11	developer	Bisa	Memilah pengembang dari aplikasi	<i>Filter</i> ini tergabung dalam pencarian kata kunci <i>List</i>
12	developer Email	Tidak bisa	Alamat email milik pengembang aplikasi	
13	developer Website	Tidak bisa	Alamat situs milik pengembang aplikasi	
14	update	Tidak bisa	Berbentuk tanggal yang selalu berubah	
15	genre	Bisa	Memilah kategori dari aplikasi	<i>List</i>
16	genreId	Bisa	Id pada	<i>Filter</i> ini

			googleplay untuk kategori aplikasi	tergabung dengan genre, isi dari data ini sama dengan isi pada atribut genre
17	description	Tidak bisa	Deskripsi/gamb aran menyeluruh dari aplikasi	
18	description HTML	Tidak bisa	Deskripsi/gamb aran menyeluruh dari aplikasi versi HTML	
19	offersIAP	Bisa	Aplikasi menyediakan pembelian di dalam aplikasi (In App Purchase) (true/false)	<i>List</i>
20	adSupport ed	Bisa	Terdapat iklan dalam aplikasi	<i>List</i>
21	androidVe rsionText	Bisa	Versi android yang dibutuhkan agar aplikasi dapat diinstall	<i>Filter</i> ini tergabung dengan poin android Version Karena memiliki isi data yang sama
22	androidVe rsion	Bisa	Versi android yang dibutuhkan agar aplikasi dapat	<i>List</i>

			diinstall	
23	contentRating	Bisa	Memilah target usia untuk pengguna dari aplikasi	List
24	video	Tidak bisa	Link video dari aplikasi	
25	recentChanges	Tidak bisa	Gambaran perubahan yang terjadi pada versi aplikasi saat ini	
26	preregister	Bisa	Aplikasi dapat dicoba terlebih dahulu (true/false)	List
27	url	Tidak bisa	Link aplikasi dapat diunduh pada website googleplay	
28	appId	Tidak bisa	Id aplikasi yang diberikan oleh googleplay untuk aplikasi	
29	Permission	Bisa	Permisson untuk tiap aplikasi	List

Setelah penggabungan metadata, dari 16 metadata terdapat 11 metadata data yang dapat menjadi penyaringan. Berikut metadata yang dapat dijadikan penyaringan:

Tabel 6.2 Hasil meta data menjadi penyaringan

No	Nama kolom	Bergabung dengan	Hasil <i>filter</i>
1	price	free	<i>Filter Harga</i>
2	minInstalls	maxInstalls	<i>Filter banyak installasi oleh user</i>
3	score		<i>Filter rating</i>
4	reviews		<i>Filter jumlah review</i>
5	genre	genreId	<i>Filter kategori</i>
6	offersIAP		<i>Filter bisa microtransac tion</i>
7	adSupported		<i>Filter iklan</i>
8	androidVersion	androidVersionText	<i>Filter versi android</i>
9	contentRating		<i>Filter untuk usia</i>
10	preregister		<i>Filter early akses</i>
11	permission		<i>Filter permission aplikasi</i>

Pada penelitian ini dihasilkan penyaringan berdasarkan 11 metadata yang ada.

6.2. Hasil perbaikan usabilitas fitur pengelolaan

Perbaikan usabilitas fitur yang terimplementasi adalah pemberian nama *template* secara otomatis.

Mencari dengan template:

Pilih template pencarian

Cari

Atau

Mencari dengan Kata kunci:

pokemon

Cari

Filter yang telah dipilih:
Everyone
Entertainment

*anda dapat menyimpan pencarian ini menjadi template ke basis data kami

Masukan judul untuk menyimpan pencarian menjadi template:

pokemon - Everyone - Entertainment

simpan

Camp Pokémon
The Pokémon Company
Rating: 4.00 Free

Pokémon GO
Niantic, Inc.
Rating: 4.10 Free

Gambar 6.3 Hasil pemberian nama *template* secara otomatis

Kolom penyimpanan yang sebelumnya terletak dibawah penyaringan dipindah bagian atas halaman hasil pencarian. Kemudian nilai yang sebelumnya kosong menjadi terisi secara otomatis berdasarkan kata kunci pencari dan filter yang telah dipilih. Sebagai contoh pada Gambar 6.3 pada kolom penyimpanan judul pencarian telah terisi Pokemon – Everyone – Entertainment yang diambil dari kata kunci dan penyaringan yang dipilih (dapat dilihat pada pojok kiri bawah Gambar 6.3).

6.3. Hasil uji usabilitas

Uji usabilitas digunakan untuk mengetahui apakah fitur yang diusulkan akan membantu pelanggan dalam mencari produk yang sesuai dengan kebutuhan dirinya dengan bantuan penyaringan. Uji usabilitas dilakukan tanggal 14 Juni 2017 sebanyak 10 responden dengan kondisi sistem telah terbentuk dan siap dijalankan (Apache + MySQL).

Beberapa evaluasi yang diperlukan dijabarkan pada tabel berikut yang akan dijadikan sebagai *backlog* untuk iterasi berikutnya.

Tabel 6.3 Backlog dari pengujian aplikasi

ID	Backlog
----	---------

B001	Memperbaiki penempatan sistem <i>template</i> pencarian pada antarmuka dan menyimpan secara otomatis
B002	Menambahkan <i>auto submit</i> pada saat memilih <i>template</i> pencarian dan <i>checkbox</i> agar ketika terpilih penyaringan bisa langsung terproses sistem
B003	Menambahkan <i>alert</i> ketika proses berhasil maupun ketika gagal.

Usabilitas testing dilakukan dengan menggunakan skenario *Test Case* yang dapat dilihat pada Lampiran 4. Partisipan diminta untuk menjalankan skenario *Test Case* yang terdiri dari 7 poin sementara penguji (dalam penelitian ini adalah penulis) memantau partisipan saat melakukan pengujian.

Tabel 6.4 *Test Case* pengujian aplikasi

Kode	Nama
TC01	Memilih <i>template</i> pencarian
TC02	Mencari produk aplikasi
TC03	Melihat hasil pencarian
TC04	Memasukan kata pencarian
TC05	Menambahkan <i>filter</i>
TC06	Mengubah <i>filter</i>
TC07	Menyimpan <i>filter</i> dan katakunci pencarian menjadi <i>template</i>

Pada akhir setiap sesi *usability testing* (mengerjakan *test case*) yang dilakukan, responden diminta untuk mengisi kuisioner *System Usability Scale* (SUS) yang digunakan untuk mengukur kemudahan dari penggunaan aplikasi. SUS adalah *tools* untuk mengukur usablitas system dengan kuesioner

terdiri dari 10 pertanyaan dengan 5 pilihan jawaban. Walau jumlah responden sangat sedikit (jumlahnya hanya 2 responden saja) SUS sudah terbukti *reliable* dan *valid* [14].

Tabel 6.5 Pernyataan dan justifikasi SUS

Pernyataan	Justifikasi
Pengguna berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin	Setuju
Pengguna berpikir sistem ini sangat rumit dan tidak perlu	Tidak Setuju
Pengguna berpikir sistem ini mudah digunakan	Setuju
Pengguna berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini	Tidak Setuju
Pengguna menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik	Setuju
Pengguna berpikir terlalu banyak inkonsistensi dari sistem ini	Tidak Setuju
Pengguna berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat	Setuju
Pengguna berpikir bahwa	Tidak Setuju

sistem ini sangat sulit untuk digunakan	
Pengguna merasa sangat percaya diri dalam menggunakan sistem ini	Setuju
Pengguna perlu belajar banyak hal sebelum dapat menggunakan sistem ini	Tidak Setuju

Adapun hasil dari kuisioner diatas dapat dilihat pada Lampiran 5 untuk lembar persetujuan (*consent form*) dan Lampiran 6 untuk kuisioner mengenai *System Usability Scale* (SUS).

Setiap jawaban kemudian diberikan penilaian. Penilaian SUS dilakukan dengan cara; untuk pertanyaan nomor ganjil skor pilihan responden dikurangi 1 (misal responden milih **setuju** yang memiliki skor 4, maka $4 - 1$), dan untuk pertanyaan nomor genap cara menghitungnya adalah 5 dikurangi skor pilihan responden (misal responden milih **tidak setuju** yang memiliki skor 2, maka $5 - 2$). Kemudian skor dari pertanyaan nomor 1 sampai 10 ditotal dan dikali 2,5.

Hasil dari *System Usability Scale* yang diberikan kepada 10 partisipan menghasilkan kemudahan dalam penggunaan sebesar 71,25% sebagaimana yang terlihat pada Gambar 6.4.

Peserta	Jenis Kelamin	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
Fitri Larasati	Perempuan	4	2	4	1	4	3	5	2	4	1	80.0%
Delina Rahayu E	Perempuan	4	2	3	2	3	3	4	2	4	2	67.5%
Haikal Rusdi	Laki-laki	4	1	5	1	4	1	4	1	5	1	92.5%
Ervi Ritya Zulvima	Perempuan	4	2	5	4	3	2	5	2	5	1	77.5%
Marina Safitri	Perempuan	4	1	5	2	5	2	4	2	5	1	87.5%
Dhamar Bagas Wisesa	Laki-laki	3	2	4	3	2	3	4	3	4	3	57.5%
Sasongko Kurniawan	Laki-laki	4	2	4	2	3	2	4	2	4	2	72.5%
Robbigh Faubendri	Laki-laki	3	4	2	4	2	2	4	2	4	2	52.5%
Yusuf Fauzan	Laki-laki	3	2	4	3	2	3	4	3	4	3	57.5%
Ramadhan Pratama Budiman	Laki-laki	3	2	4	4	5	2	4	1	4	4	67.5%
Total											71.25%	

Gambar 6.4 Hasil rekapitulasi SUS

Hasil perhitungan untuk tiap partisipan dapat dilihat pada tabel berikut:

Tabel 6.6 Perhitungan skor pengujian usabilitas

Fitri	$(4-1)+(5-2)+(4-1)+(5-1)+(4-1)+(5-3)+(5-1)+(5-2)+(4-1)+(5-1) = 32 \times 2,5 = 80$
Delina	$(4-1)+(5-2)+(3-1)+(5-2)+(3-1)+(5-3)+(4-1)+(5-2)+(4-1)+(5-2) = 27 \times 2,5 = 67,5$
Haikal	$(4-1)+(5-1)+(5-1)+(5-1)+(4-1)+(5-1)+(4-1)+(5-1)+(5-1)+(5-1) = 37 \times 2,5 = 92,5$
Ervi	$(4-1)+(5-2)+(5-1)+(5-4)+(3-1)+(5-2)+(5-1)+(5-2)+(5-1)+(5-1) = 32 \times 2,5 = 80$
Marina	$(4-1)+(5-1)+(5-1)+(5-2)+(5-1)+(5-2)+(4-1)+(5-2)+(5-1)+(5-1) = 35 \times 2,5 = 87,5$
Dhamar	$(3-1)+(5-2)+(4-1)+(5-3)+(2-1)+(5-3)+(4-1)+(5-3)+(4-1)+(5-3) = 23 \times 2,5 = 57,5$
Sasongko	$(4-1)+(5-2)+(4-1)+(5-2)+(3-1)+(5-2)+(4-1)+(5-2)+(4-1)+(5-2) = 29 \times 2,5 = 72,5$
Robbigh	$(3-1)+(5-4)+(2-1)+(5-4)+(2-1)+(5-2)+(4-1)+(5-2)+(4-1)+(5-2) = 21 \times 2,5 = 52,5$
Yusuf	$(3-1)+(5-2)+(4-1)+(5-3)+(2-1)+(5-3)+(4-1)+(5-3)+(4-1)+(5-3) = 23 \times 2,5 = 57,5$
Ramadhan	$(3-1)+(5-2)+(4-1)+(5-4)+(5-1)+(5-2)+(4-1)+(5-1)+(4-1)+(5-4) = 27 \times 2,5 = 67,5$

BAB VII

KESIMPULAN DAN SARAN

Pada bagian ini akan menjelaskan mengenai kesimpulan dari hasil penelitian dan saran untuk pengembangan penelitian di kemudian hari.

7.1. Kesimpulan

Berdasarkan pengerjaan tugas akhir dengan judul “Rancang Bangun Fitur Pengelolaan Penyaringan Produk Pada Situs Penjualan Produk Perangkat Lunak Untuk Mempermudah Pencarian” yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi pencarian dan penyaringan produk pada Google Play telah dibuat berdasarkan kebutuhan yang didapat dari literatur dan wawancara yaitu:
 - a. Sistem menyediakan pencarian yang dapat menambahkan penyaringan.
 - b. Sistem menyediakan penyimpanan pencarian dan penyaringan sebagai *template*.
 - c. Sistem menyediakan pencarian menggunakan *template* untuk membantu pengguna menemukan produk yang pernah dicari oleh pengguna sebelumnya.
 - d. Pembuatan penyaringan dipilih berdasarkan tipedata yang dibentuk dari 16 metadeta yang kemudian disederhanakan menjadi 11 metadeta.
2. Aplikasi pencarian dan penyaringan telah melewati skenario uji coba untuk semua kebutuhan dengan hasil sebagai berikut:
 - a. Hasil *System Usability Scale* yang dilakukan terhadap 10 partisipan, didapatkan bahwa sistem memberi kemudahan penggunaan sebesar 71,25%.
 - b. Evaluasi dari hasil pengujian kepada 10 partisipan yaitu:

- Memperbaiki penempatan sistem *template* pencarian pada antarmuka dan menyimpan secara otomatis.
 - Menambahkan *auto submit* pada saat memilih *template* pencarian dan *checkbox* agar ketika terpilih penyaringan bisa langsung terproses sistem.
 - Menambahkan *alert* ketika proses berhasil maupun ketika gagal.
- c. Dari hasil evaluasi dibangun perbaikan usability fitur pengelolaan untuk meningkatkan kemudahan pencarian dan penyaringan.

7.2. Saran

Saran penulis untuk penelitian dan pengembangan selanjutnya adalah sebagai berikut:

1. Untuk penelitian selanjutnya dapat ditambahkan dengan penambahan fitur melihat detail aplikasi. Detail aplikasi ini digunakan untuk melihat informasi dari produk secara rinci.
2. Aplikasi juga dapat ditambahkan semacam fitur *autocomplete* dan *live search* pada kolom *input* pencarian menggunakan kata kunci. *Autocomplete* dan *live search* dapat memudahkan pengguna dalam mencari aplikasi yang diinginkan.
3. Selain penambahan *autocomplete* dan *live search*, dapat ditambahkan juga jenis penyaringan yang ada pada *google play*, seperti menyaring berdasarkan versi android maupun berdasarkan *permission* aplikasi.

DAFTAR PUSATAKA

- [1] M. Gaasterland, "What is Content Curation? And how it's useful to you and your network.," *michielgaasterland*, 2011. [Online]. Available: <http://www.michielgaasterland.com/content-marketing/what-is-content-curation-and-how-it's-useful-to-you-and-your-network/>. [Accessed: 28-Dec-2016].
- [2] D. Tunkelang, "Faceted Search," *Synth. Lect. Inf. Concepts, Retrieval, Serv.*, vol. 1, no. 1, pp. 1–80, Jan. 2009.
- [3] J. M. Hörnkvist, W. R. Loofbourrow, and E. R. Koebler, "Faceted search results," 2011.
- [4] S. Klemmer, "Human-Centered Design: an Introduction - University of California, San Diego | Coursera." [Online]. Available: <https://www.coursera.org/learn/human-computer-interaction>. [Accessed: 22-Feb-2017].
- [5] S. S. Experience, *An Introduction to Curated Commerce*. 2013.
- [6] R. Hahn *et al.*, "Faceted Wikipedia search," *Lect. Notes Bus. Inf. Process.*, vol. 47 LNBIP, pp. 1–11, 2010.
- [7] J. Callaham, "The new Google Store is your one-stop web shop for Chromebooks, Nexus devices and more," 2015. [Online]. Available: <http://www.androidcentral.com/new-google-store-your-one-stop-web-shop-chromebooks-nexus-devices-and-more>. [Accessed: 15-Feb-2017].
- [8] Y. Wirawan, "Google Play API." .
- [9] F. Liu, C. Yu, W. Meng, and A. Chowdhury, "Effective keyword search in relational databases," *Sigmod*, p. 563, 2006.
- [10] T. Nugraha, "Tutorial Dasar Laravel," 2014.
- [11] "Laravel Development: Advantages of Using Laravel PHP Frameworks for Advanced Application -

- MyTechLogy.” [Online]. Available: <https://www.mytechlogy.com/IT-blogs/12091/laravel-development-advantages-of-using-laravel-php-frameworks-for-advanced-application/#.WWcujYiGO00>. [Accessed: 13-Jul-2017].
- [12] “Prototype - Science in Context.” [Online]. Available: <http://ic.galegroup.com/ic/scic/ReferenceDetailsPage/ReferenceDetailsWindow?failOverType=&query=&prodId=SCIC&windowstate=normal&contentModules=&display-query=&mode=view&displayGroupName=Reference&limiter=&currPage=&disableHighlighting=false&displayGroups=>. [Accessed: 13-Jul-2017].
- [13] A. S. for P. Affairs, “Usability Testing,” 2013.
- [14] J. Sauro, “MeasuringU: Measuring Usability with the System Usability Scale (SUS),” *online*, 2011. [Online]. Available: <https://measuringu.com/sus/>. [Accessed: 13-Jun-2017].
- [15] T. Connolly and C. Begg, “Database Systems: A Practical Approach to Design, Implementation, and Management,” pp. 1–1425, 2005.
- [16] “About Filters and Filter Types.” [Online]. Available: <https://docs.oracle.com/middleware/bidv1221/desktop/BIDVD/GUID-82BE87C8-57D1-47EF-8BFD-AF260D70B06B.htm#BIDVD-GUID-82BE87C8-57D1-47EF-8BFD-AF260D70B06B>. [Accessed: 11-Jul-2017].
- [17] “Using Filters for Good UX.” [Online]. Available: <https://www.newfangled.com/using-filters-for-good-ux/>. [Accessed: 11-Jul-2017].

BIODATA PENULIS



Penulis lahir di Jakarta tanggal 28 Desember 1994. Penulis merupakan anak pertama dari empat bersaudara. Penulis menempuh pendidikan formal di SDIT Nurul Fikri Depok, SMPIT Nurul Fikri Depok, dan SMAIT Nurul Fikri Depok. Pada tahun 2013 penulis mengikuti SBMPTN tulis dan diterima di Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember (ITS)

Surabaya dengan NRP 5213100107. Sebagai mahasiswa penulis aktif dalam urusan akademik, non akademik maupun organisasi. Tercatat penulis pernah menjadi kepala divisi pada Departemen Riset dan Teknologi di Badan Eksekutif Mahasiswa Fakultas Informasi Teknologi Sepuluh Nopember. Selain organisasi, penulis juga aktif dalam kepanitiaan, baik panitia dalam organisasi maupun di luar organisasi. Penulis juga pernah menjalani kerja praktik di Kantor pusat PT. Pertamina Gas di Jakarta Pusat selama 1,5 bulan pada tahun 2016.

Pada akhir semester di Jurusan Sistem Informasi, penulis mengambil bidang minat Sistem Enterprise untuk tugas akhir. Untuk kepentingan penelitian, penulis juga dapat dihubungi melalui email *umar.aqsho@gmail.com*.

Halaman ini sengaja dikosongkan

LAMPIRAN 1 SOURCE CODE APLIKASI

```
Route::get('/home', 'appController@index');
Route::post('/home', 'appController@index');

Route::get('/ajax', 'appController@ajax');
Route::post('/ajax', 'appController@ajax');

Route::get('/ajaxprice', 'appController@ajaxprice');
Route::post('/ajaxprice', 'appController@ajaxprice');
```

Gambar 9.1 Routing dari sistem yang dikembangkan

```
<form class = "sidepanel" action="" method="GET" id='filtering">
<h4>Mencari dengan template:</h4>
<select name="presetdrop">
    <option value="" disabled selected>Pilih template pencarian</option>
    @foreach($tes as $r2)
        <option value="{{ $r2->title }}">{{ $r2->title }}</option>
    @endforeach
</select>
<input class="ui-button ui-widget ui-corner-all" type="submit" data-inline="true" value="Cari">
<h4>Atau</h4>
<h4>Mencari dengan Kata kunci:</h4>
<input type="text" name="submit" placeholder="Masukan kata kunci.." value="{{Input::get('submit')}}">
<input class="ui-button ui-widget ui-corner-all" type="submit" data-inline="true" value="Cari">
<br>
```

Gambar 9.2 dropdown template dan kolom *input* kata kunci

```
<div class="filtered">
<br>
*anda dapat menyimpan pencarian ini menjadi template ke basis data kami
<br>

<br>Masukan judul untuk menyimpan pencarian menjadi template: <input type="text" name="preset"
placeholder="Judul template pencarian.." value="{{Input::get('preset')}}">
<button onclick="myFunction()" type="submit" name="Savers" data-inline="true" value="Savers"> simpan </button>
</div>
<?php
}
?>
```

Gambar 9.3 kolom input judul *template* dan tombol simpan

```

<h4>Filter untuk usia:</h4>
<input type="checkbox" name="Everyone" id="age1" style="display: none" value="Everyone"
</php if(isset($_GET['Everyone'])) echo "checked='checked'"; >> <label for="age1" >Everyone</label>
<input type="checkbox" name="Teen" id="age2" style="display: none" value="Teen"
</php if(isset($_GET['Teen'])) echo "checked='checked'"; >> <label for="age2" >Teen</label>
<input type="checkbox" name="Mature" id="age3" style="display: none" value="Mature 17+*
</php if(isset($_GET['Mature'])) echo "checked='checked'"; >> <label for="age3" >Mature 17+</label>
<input type="checkbox" name="Adults" id="age4" style="display: none" value="Adults only"
</php if(isset($_GET['Adults'])) echo "checked='checked'"; >> <label for="age4" >Adults Only</label>
<input class="ui-button ui-widget ui-corner-all" type="submit" data-inline="true" value="Submit">

<h4>Filter kategori:</h4>
<input type="checkbox" name="Entertainment" id="cat1" style="display: none" value="Entertainment" >> <label for="cat1" >Entertainment</label>
<input type="checkbox" name="Audio" id="cat2" style="display: none" value="Music" >> <label for="cat2" >Music & Audio</label>
<input type="checkbox" name="Communication" id="cat3" style="display: none" value="Communication" >> <label for="cat3" >Communication</label>
<input type="checkbox" name="Social" id="cat4" style="display: none" value="Social" >> <label for="cat4" >Social</label>
<input type="checkbox" name="Action" id="cat5" style="display: none" value="Action" >> <label for="cat5" >Action</label>
<input type="checkbox" name="Tools" id="cat6" style="display: none" value="Tools" >> <label for="cat6" >Tools</label>
<input type="checkbox" name="Word" id="cat7" style="display: none" value="Word" >> <label for="cat7" >Word</label>
<input type="checkbox" name="Shopping" id="cat8" style="display: none" value="Shopping" >> <label for="cat8" >Shopping</label>
<input type="checkbox" name="Simulation" id="cat9" style="display: none" value="Simulation" >> <label for="cat9" >Simulation</label>
<input type="checkbox" name="Puzzle" id="cat10" style="display: none" value="Puzzle" >> <label for="cat10" >Puzzle</label>
<input type="checkbox" name="Arcade" id="cat11" style="display: none" value="Arcade" >> <label for="cat11" >Arcade</label>
<input type="checkbox" name="Maps" id="cat12" style="display: none" value="Maps" >> <label for="cat12" >Maps & Navigation</label>
<input type="checkbox" name="Adventure" id="cat13" style="display: none" value="Adventure" >> <label for="cat13" >Adventure</label>
<input type="checkbox" name="Sports" id="cat14" style="display: none" value="Sports" >> <label for="cat14" >Sports</label>
<input type="checkbox" name="Racing" id="cat15" style="display: none" value="Racing" >> <label for="cat15" >Racing</label>
<input type="checkbox" name="Productivity" id="cat16" style="display: none" value="Productivity" >> <label for="cat16" >Productivity</label>
<input type="checkbox" name="Casual" id="cat17" style="display: none" value="Casual" >> <label for="cat17" >Casual</label>
<input type="checkbox" name="Board" id="cat18" style="display: none" value="Board" >> <label for="cat18" >Board</label>
<input type="checkbox" name="Personalization" id="cat19" style="display: none" value="Personalization" >> <label for="cat19" >Personalization</label>
<input type="checkbox" name="Strategy" id="cat20" style="display: none" value="Strategy" >> <label for="cat20" >Strategy</label>
<input type="checkbox" name="Photography" id="cat21" style="display: none" value="Photography" >> <label for="cat21" >Photography</label>
<input type="checkbox" name="Music" id="cat22" style="display: none" value="Music" >> <label for="cat22" >Music</label>
<input type="checkbox" name="News" id="cat23" style="display: none" value="News" >> <label for="cat23" >News & Magazines</label>
<input type="checkbox" name="Finance" id="cat24" style="display: none" value="Finance" >> <label for="cat24" >Finance</label>
<input type="checkbox" name="Card" id="cat25" style="display: none" value="Card" >> <label for="cat25" >Card</label>
<input type="checkbox" name="Travel" id="cat26" style="display: none" value="Travel" >> <label for="cat26" >Travel & Local</label>
<input type="checkbox" name="Food" id="cat27" style="display: none" value="Food" >> <label for="cat27" >Food & Drink</label>
<input type="checkbox" name="Lifestyle" id="cat28" style="display: none" value="Lifestyle" >> <label for="cat28" >Lifestyle</label>
<input type="checkbox" name="Weather" id="cat29" style="display: none" value="Weather" >> <label for="cat29" >Weather</label>

```

Gambar 9.4 filter checkbox yang tersedia pada halaman pencarian

```

$(window).load(function(){
    $("#score-max").on("slidestop", function(e){
        var min = document.getElementById('score-min').value;
        var max = document.getElementById('score-max').value;

        $.ajax({
            url: 'ajax',
            type: 'GET',
            data: {"score-min":min,"score-max":max},
            success: function(response){
                // alert(response);
                $("#tempel").html(response);
            }
        });
    });
    $("#score-min").on("slidestop", function(e){
        var min = document.getElementById('score-min').value;
        var max = document.getElementById('score-max').value;

        $.ajax({
            url: 'ajax',
            type: 'GET',
            data: {"score-min":min,"score-max":max},
            success: function(response){
                // alert(response);
                $("#tempel").html(response);
            }
        });
    });
    $("#price-min").on("slidestop", function(e){
        var min = document.getElementById('price-min').value;
        var max = document.getElementById('price-max').value;

        $.ajax({
            url: 'ajaxprice',
            type: 'GET',
            data: {"price-min":min,"price-max":max},
            success: function(response){
                // alert(response);
                $("#tempel").html(response);
            }
        });
    });
});

```

Gambar 9.5 *filter range slider* yang tersedia pada halaman pencarian

```

foreach ($database as $r1) {
    echo "<div class='panel'>";
    echo "<div class='icon'>";
    echo "<img width='130' src='\".$r1->icon.\"'> ";
    echo "</div>";
    echo "<div class='title'>";
    echo "\". $r1->title;";
    echo "<span class='paragraph-end'></span>";
    echo "</div>";
    echo "<div class='devname'>";
    echo "\". $r1->developer;";
    echo "<span class='paragraph-end'></span>";
    echo "</div>";
    echo "<br>";
    echo "<div class='score'>";
    echo " Rating: ".$r1->score;
    echo "</div>";
    echo "<div class='price'>";
    if($r1->price != '0'){
        echo "\".$r1->price;";
    }else{
        echo "Free";
    }
    echo "</div>";
    echo "</div>";
}
?>
</div>
<?php
if (!isset($_GET['submit']) AND !isset($_GET['presetttitle']))){
?>
    <br>
    <center>
    <div class="pagination">
        {{ $database->links() }}
    </div>
    </center>
<?php
}else{
}
}

```

Gambar 9.6 Menampilkan hasil pencarian


```

public function ajax(){
    $scoremin = $_GET['score-min'];
    $scoremax = $_GET['score-max'];

    $database = DB::select("SELECT ap.icon, ap.title, ap.price, ap.score, ap.developer FROM applications ap
    LEFT JOIN applications_facts af ON af.app_id=ap.id
    WHERE (af.facet_id = 8 AND (af.value BETWEEN $scoremin AND $scoremax)) ");

    foreach ($database as $r1) {=
}

```

Gambar 9.7 Query filter rating

```

public function ajaxprice(){
    $pmin = $_GET["price-min"];
    if($pmin == '0'){
        $pricemin = $_GET["price-min"];
        $pricemax = $_GET["price-max"];
    }else{
        $pricemin = '$'.$_GET["price-min"];
        $pricemax = '$'.$_GET["price-max"];
    }

    $database = DB::select("SELECT ap.icon, ap.title, ap.price, ap.score, ap.developer FROM applications ap
    LEFT JOIN applications_facts af ON af.app_id=ap.id
    WHERE (af.facet_id = 4 AND (af.value BETWEEN '".$pricemin."' AND '".$pricemax."')) ");

    foreach ($database as $r1) {=
}

```

Gambar 9.8 Query filter harga

```

if(!empty($_GET['presetdrop'])){
    $presetdrop = $_GET['presetdrop'];
    $resdrop = DB::select("SELECT * FROM filter_preset WHERE title LIKE '%$presetdrop%'");
    $database = DB::select("SELECT * FROM filter_preset WHERE title LIKE '%$presetdrop%'");
    $tes = DB::table("filter_preset")->get();
    return view('home', ['database' => $database, 'tes' => $tes, 'resdrop' => $resdrop]);
}

```

Gambar 9.9 Query pencarian dengan template dropdown

```

$database = DB::select("SELECT ap.icon, ap.title, ap.price, ap.score, ap.developer FROM applications ap
LEFT JOIN applications_facts af ON af.app_id=ap.id
WHERE (ap.title LIKE '%$term%' OR ap.developer LIKE '%$term%')
AND ($age OR $genre)
GROUP BY ap.id, ap.icon, ap.title, ap.price, ap.score, ap.developer");

$tes = DB::table("filter_preset")->get();
return view('home', ['database' => $database, 'tes' => $tes]);

```

Gambar 9.10 Query pencarian dengan kata kunci dan filter checkbox

LAMPIRAN 2 USABILITY TEST PLAN DASHBOARD

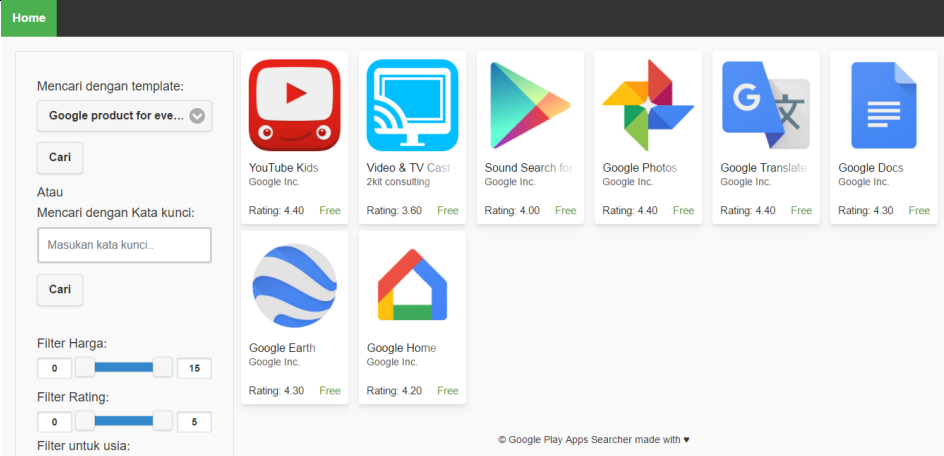
AUTHOR		CONTACT DETAILS		FINAL DATE FOR COMMENTS	
UMAR AL AQSHO		umar.aqsho@gmail.com			
PRODUCT UNDER TEST What's being tested? What are the business and experience goals of the product? Website pencarian dan penyaringan produk google play. tujuannya memudahkan mencari produk yang diinginkan	TEST OBJECTIVES What are the goals of the usability test? What specific questions will be answered? What hypotheses will be tested? Dapatkah anda langsung menemukan aplikasi yang diinginkan? Masalah apa saja yang ada ketika mencari produk di Google play? Fitur apa saja yang sebaiknya ada dalam mencari produk aplikasi agar mudah ditemukan?	PARTICIPANTS How many participants will be recruited? What are their key characteristics? Pengguna perangkat android yang menggunakan Google Play	TEST TASKS What are the test tasks? Memilih template pencarian Mencari produk aplikasi Melihat hasil pencarian Memasukan kata pencarian Menambah filter Mengubah filter Menyimpan filter dan kata kunci pencarian menjadi template	RESPONSIBILITIES Who is involved in the test and what are their responsibilities? Umar Al Aqsho (PM, Moderator)	
BUSINESS CASE Why are we doing this test? What are the benefits? What are the risks of not testing? Testing bertujuan untuk mendapatkan feedback dan masukan untuk pengembangan aplikasi selanjutnya		EQUIPMENT What equipment is required? How will you record the data? Laptop dan aplikasi yang terinstall Form responden Akses poin untuk koneksi internet		LOCATION & DATES Where and when will the test take place? When and how will the results be shared? 21 Maret 2017. Lokasi di Mulyosari. dapat diakses di http://localhost/home	
PROCEDURE What are the main steps in the test procedure? <div> <div>Perkenalan identitas diri dan tujuan testing</div> <div>penjelasan lingkup aplikasi dan interview</div> <div>Pelaksanaan Test task</div> <div>responden mengisi kuesioner</div> <div>Post-test interview</div> <div>Terima kasih dan insentif ke responden</div> </div>					

LAMPIRAN 3 HASIL WAWANCARA

ID	Question and answer
QA1	Dapatkah anda langsung menemukan aplikasi yang diinginkan ketika mencarinya dengan google play?
QA2	Masalah/kekurangan apa saja yang ada ketika mencari produk aplikasi pada google play?
QA4	Fitur apa saja yang sebaiknya ada dalam mencari produk aplikasi agar mudah ditemukan?
QA5	Apakah pencarian dalam google play dapat dikembangkan menjadi lebih baik?

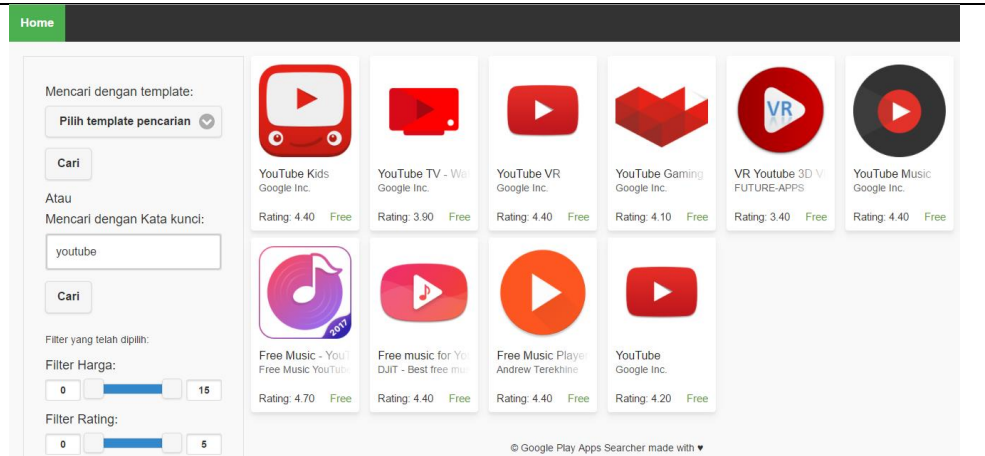
LAMPIRAN 4 TEST CASE

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC01	Memilih <i>Template</i> pencarian	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Mencari produk aplikasi dengan menggunakan <i>dropdown template</i>			
Step	1. Klik <i>dropdown</i> 2. Pilih salah satu <i>template</i> 3. Tekan tombol cari			
Input	Judul template pencarian			
Hasil dan Ekspektasi	1. Sistem akan menampilkan produk pencarian yang kata kunci dan penyaringannya tersimpan dalam <i>template</i> pada halaman hasil pencarian			

Hasil Testing	<p>1. Sistem menampilkan produk pencarian yang kata kunci dan penyaringannya tersimpan dalam <i>template</i> pada halaman hasil pencarian</p>
Bukti	 <p>The screenshot displays the Google Play Apps Searcher interface. At the top, there is a 'Home' tab. Below it, the search section includes a 'Mencari dengan template:' dropdown set to 'Google product for eve...', a 'Cari' button, and an 'Atau' section with 'Mencari dengan Kata kunci:' and a text input field 'Masukan kata kunci...' followed by another 'Cari' button. Filter options include 'Filter Harga:' with a slider from 0 to 15, 'Filter Rating:' with a slider from 0 to 5, and 'Filter untuk usia:'. The main area shows a grid of app cards for YouTube Kids, Video & TV Cast, Sound Search for, Google Photos, Google Translate, Google Docs, Google Earth, and Google Home. Each card displays the app icon, name, developer, rating, and price (all are 'Free'). At the bottom right, it says '© Google Play Apps Searcher made with ▼'.</p>

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC02	Mencari Produk aplikasi	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Mencari produk aplikasi dengan menggunakan kolom kata kunci			
Step	1. Masukan kata kunci pada kolom pencarian 2. Tekan tombol cari			
Input	Kata kunci pencarian			
Hasil dan Ekspektasi	1. Sistem akan menampilkan produk pencarian yang memiliki kata kunci yang dicari			
Hasil Testing	1. Sistem menampilkan produk pencarian yang memiliki kata kunci yang dicari			

Bukti



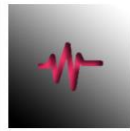
Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC03	Melihat hasil pencarian	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Mendapatkan hasil produk aplikasi			
Step	1. Mencari dengan <i>template</i> /kata kunci 2. Tekan tombol cari 3. Melihat hasil pencarian			
Input	Pencarian dari <i>template</i> /kata kunci			
Hasil dan Ekspektasi	1. Sistem akan menampilkan hasil pencarian			
Hasil Testing	1. Sistem menampilkan hasil pencarian			

Bukti



Piano Music Tiles
Piano Music House

Rating: 4.50 Free



Famous on Music
FHX Express

Rating: 2.20 Free



Free Pandora Music
smith lab

Rating: 4.30 Free



Spotify Music
Spotify Ltd.

Rating: 4.50 Free



Free Music
Free Music - Enjoy

Rating: 4.40 Free



YouTube Music
Google Inc.

Rating: 4.40 Free



SoundCloud - Music
SoundCloud

Rating: 4.40 Free



iHeartRadio Free
iHeartMedia, Inc.

Rating: 4.70 Free



Free Music - YouTube
Free Music YouTube

Rating: 4.70 Free



Shazam - Discover
Shazam Entertainment

Rating: 4.40 Free



TuneIn Radio - Real
TuneIn Inc

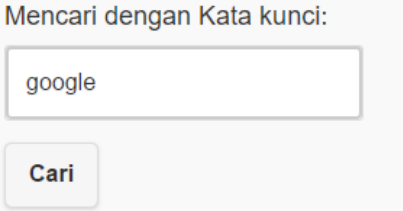
Rating: 4.40 Free



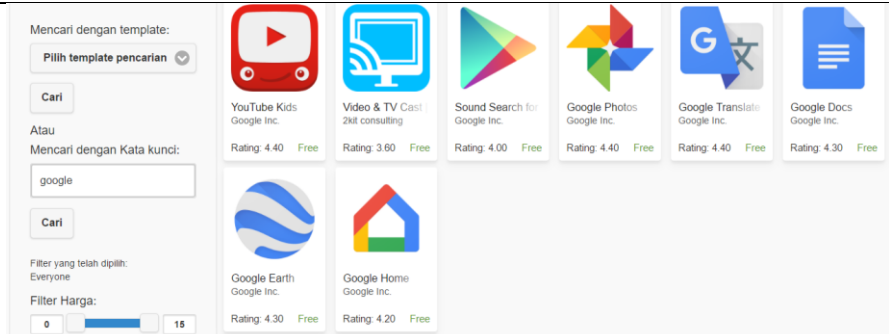
Apple Music
Apple Inc.

Rating: 3.40 Free

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC04	Memasukan kata pencarian	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Memasukan kata kunci di kolom pencarian			
Step	1. Memasukan kata kunci di kolom pencarian			
Input	Kata kunci			
Hasil dan Ekspektasi	1. Sistem akan memasukan kata kunci yang diketik			
Hasil Testing	1. Sistem memasukan kata kunci yang diketik			

Bukti	
-------	--

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC05	Menambahkan filter	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Menambahkan filter pada pencarian yang dilakukan			
Step	1. Memasukan kata kunci di kolom pencarian 2. Menambahkan salah satu/lebih penyaringan			

	3. Tekan tombol submit/cari
Input	Kata kunci, penyaringan
Hasil dan Ekspektasi	1. Sistem akan menampilkan produk pencarian yang memiliki kata kunci yang sesuai dengan penyaringan
Hasil Testing	1. Sistem menampilkan produk pencarian yang memiliki kata kunci yang sesuai dengan penyaringan
Bukti	 <p>The screenshot displays a Google search interface. On the left, there's a search bar with 'google' entered and a 'Cari' button. Below it, a section titled 'Mencari dengan template:' shows a dropdown menu for 'Pilih template pencarian'. Another 'Cari' button is present. Below that, a section titled 'Atau' shows a search bar with 'google' and another 'Cari' button. Further down, a section titled 'Mencari dengan Kata kunci:' shows a search bar with 'google' and a 'Cari' button. Below that, a section titled 'Filter yang telah dipilih:' shows 'Everyone' selected. A section titled 'Filter Harga:' shows a price range from 0 to 15. On the right, there's a grid of search results for Google products, each with a rating and a 'Free' label. The products shown are YouTube Kids (Rating: 4.40), Video & TV Cast (Rating: 3.60), Sound Search for (Rating: 4.00), Google Photos (Rating: 4.40), Google Translate (Rating: 4.40), Google Docs (Rating: 4.30), Google Earth (Rating: 4.30), and Google Home (Rating: 4.20).</p>

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC06	Mengubah filter	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Mengubah filter pada pencarian yang dilakukan			
Step	1. Melihat hasil pencarian dengan tambahan <i>filter</i> 2. Mengubah <i>filter</i> yang telah dipilih 3. Tekan tombol submit/cari			
Input	Kata kunci, penyaringan			
Hasil dan Ekspektasi	1. Sistem akan menampilkan produk pencarian yang memiliki kata kunci yang sesuai dan berbeda dengan penyaringan sebelumnya			
Hasil Testing	1. Sistem menampilkan produk pencarian yang memiliki kata kunci yang sesuai dan berbeda dengan penyaringan sebelumnya			

Bukti

Mencari dengan template:

Pilih template pencarian ▼

Cari

Atau

Mencari dengan Kata kunci:

google

Cari

Filter yang telah dipilih:
Teen

Filter Harga:

0

15



Google Play Games
Google Inc.

Rating: 4.20 Free



YouTube Kids
Google Inc.

Rating: 4.40 Free



YouTube TV - Watch
Google Inc.

Rating: 3.90 Free



YouTube VR
Google Inc.

Rating: 4.40 Free



YouTube Gaming
Google Inc.

Rating: 4.10 Free



YouTube Music
Google Inc.

Rating: 4.40 Free



Google Play Music
Google Inc.

Rating: 3.90 Free



Sound Search for
Google Inc.

Rating: 4.00 Free



Google Photos
Google Inc.

Rating: 4.40 Free



Google Translate
Google Inc.

Rating: 4.40 Free

Test Case Design Info		Test Case Execution Info		
Test Case	Use Case	Sukses/gagal	Tanggal/Waktu	Tester
TC06	Mengubah filter	Sukses	14 Juni 2017	Partisipan
			15:00 WIB	
Objective	Mengubah filter pada pencarian yang dilakukan			
Step	1. Melihat hasil pencarian dengan tambahan <i>filter</i> 2. Menambahkan judul <i>template</i> 3. Menekan tombol simpan.			
Input	Kata kunci, penyaringan, judul <i>template</i>			
Hasil dan Ekspektasi	1. Sistem akan menyimpan pencarian dan penyaringan yang dilakukan ke basis data			
Hasil Testing	1.			

Bukti	<p>*anda dapat menyimpan pencarian ini menjadi template ke basis data kami</p> <p>Masukan judul untuk menyimpan pencarian menjadi template:</p> <div data-bbox="743 367 1139 451">Google product for photograp </div> <div data-bbox="743 474 885 546">simpan</div>
-------	---

LAMPIRAN 5 CONSENT FORM

PENDAHULUAN

Testing ini dilakukan untuk melakukan penilaian terhadap kegunaan (Usabilitas) dari aplikasi yang akan dibuat oleh peneliti.

Aplikasi yang akan dibuat diberi nama Google Play searcher. Google Play searcher merupakan sistem pencarian dan penyaringan produk aplikasi yang berada pada Google Play Store.

Pengembangan aplikasi Google Play searcher bertujuan, yaitu untuk membantu pengguna untuk lebih mudah dalam mencari produk aplikasi di Google Play.

Usability testing ini akan memakan waktu sekitar 10 menit dan media yang digunakan adalah aplikasi *Minimum Viable Product*. Harapannya hasil masukan dari *testing* akan dibuat untuk mengembangkan aplikasi kedepannya. seluruh fitur dan fungsi dari aplikasi dibuat berdasarkan literatur dan wawancara dengan pihak yang terkait dengan penelitian ini.

Terdapat 2 Bagian pertanyaan yang harus dijawab. Pada bagian pertama menggunakan tipe soal likert dengan skala 1 – 5 yang bertujuan mengukur usabilitas dari aplikasi. Pada bagian kedua menggunakan tipe soal bebas, untuk mengambil evaluasi dan saran untuk pengembangan aplikasi.

Terima kasih atas partisipasi anda.

FORM PERSETUJUAN

Saya mengerti bahwa partisipasi dalam studi kegunaan ini bersifat sukarela.

Silakan isi form berikut untuk menunjukkan bahwa Anda telah membaca dan Anda memahami informasi pada formulir ini dan bahwa setiap pertanyaan yang mungkin Anda miliki tentang sesi telah dijawab.

Tanggal : _____

Nama Lengkap : _____

Jenis Kelamin : _____

Usia : _____

Tanda tangan : _____

LAMPIRAN 6 SYSTEM USABILITY SCALE FORM

Instruksi : Isikan setiap jawaban pernyataan dengan memberikan tanda *checklist* (√) atau *cross* (x) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.

1 = Sangat tidak setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat setuju

Pernyataan	1	2	3	4	5
Pengguna berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin					
Pengguna berpikir sistem ini sangat rumit dan tidak perlu					
Pengguna berpikir sistem ini mudah digunakan					
Penggguna berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini					
Pengguna menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik					
Pengguna berpikir terlalu banyak inkonsistensi dari sistem ini					
Pengguna berpikir banyak orang yang akan dapat belajar menggunakan sistem					

ini dengan cepat					
Pengguna berpikir bahwa sistem ini sangat sulit untuk digunakan					
Pengguna merasa sangat percaya diri dalam menggunakan sistem ini					
Pengguna perlu belajar banyak hal sebelum dapat menggunakan sistem ini					

BAGIAN II EVALUASI

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? sebutkan

2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan

3. Saran anda untuk perbaikan kedepannya?
